ROBUST NETWORK DESIGN UNDER POLYHEDRAL TRAFFIC UNCERTAINTY

A DISSERTATION SUBMITTED TO
THE DEPARTMENT OF INDUSTRIAL ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Ayşegül ALTIN
July, 2007

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Mustafa Ç. Pınar (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Prof. Dr. Levent Kandiller

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Oya Ekin Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assoc. Prof. Ezhan Karaşan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of doctor of philosophy.

Assist. Prof. Hande Yaman

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet B. Baray
Director of the Institute

ABSTRACT

ROBUST NETWORK DESIGN UNDER POLYHEDRAL TRAFFIC UNCERTAINTY

Ayşegül ALTIN
Ph.D. in Industrial Engineering
Supervisor: Prof. Dr. Mustafa Ç. Pınar
July, 2007

In this thesis, we study the design of networks robust to changes in demand estimates. We consider the case where the set of feasible demands is defined by an arbitrary polyhedron. Our motivation is to determine link capacity or routing configurations, which remain feasible for any realization in the corresponding demand polyhedron. We consider three well-known problems under polyhedral demand uncertainty all of which are posed as semi-infinite mixed integer programming problems. We develop explicit, compact formulations for all three problems as well as alternative formulations and exact solution methods.

The first problem arises in the Virtual Private Network (VPN) design field. We present compact linear mixed-integer programming formulations for the problem with the classical hose traffic model and for a new, less conservative, robust variant relying on accessible traffic statistics. Although we can solve these formulations for medium-to-large instances in reasonable times using off-the-shelf MIP solvers, we develop a combined branch-and-price and cutting plane algorithm to handle larger instances. We also provide an extensive discussion of our numerical results.

Next, we study the Open Shortest Path First (OSPF) routing enhanced with traffic engineering tools under general demand uncertainty with the motivation to discuss if OSPF could be made comparable to the general unconstrained routing (MPLS) when it is provided with a less restrictive operating environment. To the best of our knowledge, these two routing mechanisms are compared for the first time under such a general setting. We provide compact formulations for both routing types and show that MPLS routing for polyhedral demands can be computed in polynomial time. Moreover, we present a specialized branch-and-price algorithm strengthened with the inclusion of cuts as an exact solution

tool. Subsequently, we compare the new and more flexible OSPF routing with MPLS as well as the traditional OSPF on several network instances. We observe that the management tools we use in OSPF make it significantly better than the generic OSPF. Moreover, we show that OSPF performance can get closer to that of MPLS in some cases.

Finally, we consider the Network Loading Problem (NLP) under a polyhedral uncertainty description of traffic demands. After giving a compact multi-commodity formulation of the problem, we prove an unexpected decomposition property obtained from projecting out the flow variables, considerably simplifying the resulting polyhedral analysis and computations by doing away with metric inequalities, an attendant feature of most successful algorithms on NLP. Under the hose model of feasible demands, we study the polyhedral aspects of NLP, used as the basis of an efficient branch-and-cut algorithm supported by a simple heuristic for generating upper bounds. We provide the results of extensive computational experiments on well-known network design instances.

Keywords: Robust network design, polyhedral traffic uncertainty, Virtual Private Network, Open Shortest Path First, Network Loading Problem, Branch-and-Price.

ÖZET

ÇOKYÜZLÜ TRAFİK BELİRSİZLİĞİ DURUMUNDA GÜRBÜZ AĞ TASARIMI

Ayşegül ALTIN
Endüstri Mühendisliği, Doktora
Tez Yöneticisi: Prof. Dr. Mustafa Ç. Pınar
Temmuz, 2007

Bu tez çalışmasında talep tahminlerindeki değişikliklere karşı dayanıklı olan ağların tasarımı incelenmiştir. Olurlu talepler kümesinin gelişigüzel bir çokyüzlü ile tanımlandığı durum dikkate alınmıştır. Çalışmanın amacı, bir çokyüzlüye ait bütün gerçekleşmeler için geçerliliği devam eden gürbüz ayrıt kapasitesi ve yol atama yapılandırmalarını belirlemektir. Aslen yarı-sonsuz karışık tamsayı programlama modelleri ile tanımlanan ve çok bilinen üç problem üzerinde çalışılmıştır. Bu problemler için açık ve tıkız formülasyonlara ek olarak alternatif formülasyonlar ve kesin çözüm yöntemleri geliştirilmiştir.

Dikkate alınan ilk problem Zahiri Özel Ağ tasarımı ile ilgilidir. Hem klasik hortum modeli hem de rahat ulaşılabilir trafik istatistiklerini kullanan, yeni ve daha az temkinli gürbüz bir talep modeli için problemin tıkız doğrusal karışık tamsayı programlama formülasyonları önerilmiştir. Bu modeller, orta ve büyük ölçekli örnekler için mevcut ticari çözücüler kullanılarak çözülebiliyorken daha büyük örneklerde kullanmak amacıyla birleşik bir dal-fiyat ve kesme yüzeyi algoritması geliştirilmiştir. Ayrıca, elde edilen deneysel sonuçların kapsamlı bir analizi de sunulmaktadır.

İncelenen ikinci problem, genel talep belirsizliği durumunda trafik mühendisliği gereçleri ile geliştirilmiş esnek En Kısa Yol Öncelikli Güzergah Atama (OSPF) problemidir. Bu konuda hedeflenen, yeterli esnekliğin sağlanması durumunda OSPF mekanizmasının MPLS gibi serbest yol atama yöntemleri kadar yüksek bir performans gösterip gösteremeyeceğini incelemektir. Sözkonusu yol atama tekniklerinin bu kadar genel bir durum için karşılaştırıldığı daha başka bir çalışma bilinmemektedir. Tezin bu bölümünde, öncelikle herhangi bir çokyüzlü ile her iki mekanizma için tıkız formülasyonlar verilmektedir. Ayrıca, çokyüzlü talep tanımı için en iyi MPLS yol yapılandırmasının polinom zamanda hesaplanabileceği

gösterilmektedir. Daha sonra ise kesin çözüm yöntemi olarak kullanılacak ve kesme yüzeyleri ile güçlendirilmiş bir dal-fiyat algoritması tanıtılmaktadır. Yeni geliştirilen esnek OSPF mekanizması, bu modeller ve çözüm yöntemi kullanılarak MPLS'nin yanısıra mevcut OSPF tekniği ile de karşılaştırılmaktadır. Yeni durumda, kullanılan yönetim gereçleri sayesinde, mevcut OSPF mekanizmasının önemli derecede geliştirilebileceği gösterilmiştir. Diğer yandan, bazı durumlarda MPLS'nin ve esnek OSPF'nin performanslarının kıyaslanabilir olduğu da gözlenmiştir.

Tezde son olarak çokyüzlü trafik talep belirsizliği ile Ağ Yükleme Problemi (NLP) ele alınmaktadır. Problemin, çok ürünlü durum için, tıkız bir formülasyonu verildikten sonra olurlu çözüm kümesinin bir alt uzaydaki izdüşümü incelenmektedir. Bu sayede mevcut en iyi NLP algoritmalarında bile kullanılmak zorunda kalınan ölçev eşitsizlikleri bertaraf edilmiştir. Neticesinde doğan çokyüzlünün analizi ve problemin çözümü önemli derecede kolaylaşmıştır. Bu suretle, olurlu trafik taleplerinin hortum modeli ile tanımlandığı durum için problem çokyüzlüsünün özellikleri incelenmiştir. Elde edilen özellikler, en iyi çözüm değeri için üst sınırları hesaplayan basit bir buluşsal ile kuvvetlendirilmiş bir dal-kesi yönteminde kullanılmıştır. Son olarak, iyi bilinen ağ tasarımı örnekleri için kapsamlı deney sonuçları verilmiştir.

Anahtar sözcükler: Gürbüz ağ tasarımı, çokyüzlü trafik belirsizliği, Zahiri Özel Ağ, En Kısa Yol Öncelikli Yol Atama, Ağ Yükleme Problemi, Dal-Fiyat yöntemi.

To my family with deepest gratitude and inmost feelings...

Acknowledgement

By the time I have started my PhD, I have had the chance to work with many valuable researchers and numerious student colleagues whose miscellaneous contributions to this thesis merits my inmost thanks. I feel privileged to have known all these people whose names I want to mention in this acknowledgement.

Firstly, I would like to express my deepest gratitude to my supervisor Prof. Mustafa Ç. Pınar for all his guidance, tolerance, and unreserved support. He made me feel that he had every confidence in me, which has been the impulsion in all frustrating periods of the research. His discipline as a scientist has contributed greatly to my academic development. I feel extraordinarily fortunate for having him as my supervisor. Moreover, I gratefully acknowledge Pietro Belotti for all his invaluable help and substantial contribution to this thesis. I have tortured him with hundreds of questions, which he has answered with all his unfailing patience and courtesy. I believe it was a great opportunity for me to work with him.

I wish to express my heart-felt thanks to Asst. Prof. Hande Yaman, Assoc. Prof. Oya Ekin Karaşan, and Asst. Prof. Emre Alper Yıldırım, who have always been ready to provide help. Not everyone can be so successful and so modest at the same time. I am also thankful to Assoc. Prof. Ezhan Karaşan and Prof. Levent Kandiller since they accept to be members of the thesis committee, devote their valuable time to read the thesis and provide constructive comments.

It is a pleasure for me to pay tribute to Prof. Francesco Maffioli, Assoc. Prof. Edoardo Amaldi, Prof. Ali Ridha Mahjuob, and all colleagues in the optimization group in Politecnico di Milano as well as in EPOC in the Université Blaise Pascal for their hospitality and the precious academic environments they provide me during my visits.

I wish to thank Banu Yüksel Özkaya and Sinan Gürel for being so considerate and gracious. I am also thankful to Aykut Özsoy and Tolga Bektaş for their unpriced camaraderie. Furthermore, my special thanks go to Iraz Toprak Aydın,

Çiğdem Sevim, Filiz Gürtuna, and Zümbül Bulut for their sincere friendship. I would also acknowledge Önder Bulut, Ahmet Camcı, and Evren Körpeoğlu for the entertaining chats and afternoon escapades. Moreover, my thanks are due to Nurdan Ahat, Mehmet Mustafa Tanrıkulu, Muzaffer Mısırcı, Gülay Samatlı-Paç, Fazıl Paç, Esra Aybar, İpek Keleş, Seda Nalbant, Hakan Gültekin, Murat Kalaycılar, Evren Kahramanoğlu, Nasuh Çağdaş Büyükkaramıklı, Erdinç Mert, Banu Karakaya, Yahya Saleh, Onur Özkök, Sibel Alumur, Utku Koç as well as Sıtkı Gülten, Emre Uzun, Könül Bayramoğlu, Tuğçe Akbaş, Hatice Çalık, Gökay Erön, Yüce Çınar, and all other colleagues for providing such a stimulating and friendly environment to learn and work.

I express my sincere thanks to all professors in the industrial engineering department as well as Yeşim Karadeniz for being so nice over the last five years. Furthermore, I would like to extend my thanks to everybody who has somehow contributed to the successful completion of this thesis whose names I could not mention one by one personally.

Words become insufficient when it comes to my family. There is no way I can literally express my gratitude to them. I owe them a lot for every single pleasant thing I have in my life. My parents deserve special mention for their everlasting support and love. My father Prof. Murat Altın has made the most valuable effort for me to form a learning character. The PhD is much easier when you have such a great role model. My mother Nuran Altın raised me with all her patience and caring love. Her priceless advices lighten my way in every stage of my life. I also thank my sisters Funda and Fulya not only for being so tolerant toward me in all those difficult times but also for being my best friends in life. Besides, I want to mention my niece Elif Naz who is my little angel and most precious treasure.

Last but not the least, I would like to thank Mehmet Kayhan, who has enriched my life with his presence. His sympathy and persistent confidence in me was helpful to surmount many difficulties.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

Asym-G ... VPN design with Asymmetric Hose Model and General Routing
Asym-T ..... VPN design with Asymmetric Hose Model and Tree Routing
B&B ..... Branch-and-Bound
B&C ..... Branch-and-Cut
B&P ..... Branch-and-Price with cuts
BPC ..... Branch-and-Price-and-Cut
ECMP ..... Equal Cost Multi-Path Routing
HCF ..... High Capacity Facility
IGP ..... Interior Gateway Protocol
IP ..... Internet Protocol
IS-IS ..... Intermediate System to Intermediate System
LCF ..... Low Capacity Facility
MIP ..... Mixed Integer Programming
MPLS ..... Multi-Protocol Label Switching
NLP ..... Network Loading Problem
OSPF ..... Open Shortest Path First
Rob-G ..... Robust VPN design problem
RLP ..... Restricted Linear Problem
SP tree ..... Shortest Paths tree
Sym-G ..... VPN design with Symmetric Hose Model and General Routing
Sym-T ..... VPN design with Symmetric Hose Model and Tree Routing
TM ..... Traffic Matrix
VPN ..... Virtual Private Network
3-SAT ..... 3-Satisfiability Problem

Chapter 1

Introduction

A network is a structure made up of entities (nodes) and the ties between the pairs of these entities. The set of nodes and the relevant links that constitute this relational structure differs depending on the reason of its existence. For example, in a social network, we have people or organizations as the entities and the relations between them can be shaped according to various criteria such as likes/dislikes, kinship/friendship for people and trade or partnership for organizations. Obviously, we can talk about a network once there are some nodes in need of communication. This fact on its own is more than enough to understand the importance of the networks.

In this thesis, we consider the design of networks within engineering and business framework where the primary goal is to design an operational network that maintains reliable communication between its nodes. Although there are many different types of such networks used in various business environments, they are all based on the transfer of work elements (commodities), which are necessary for the continuation of the system forming the network. For example, in a supply chain network of a company, nodes can be the factories, warehouses, retailers, and end-users whereas links could correspond to direct shipment of products from one node to the other. Another example is the telecommunication networks where data packets are carried on the links and through the nodes. Their popularity is increasing in parallel to the growing share of information technology

in today's business applications. Some examples of telecommunication networks are Internet, computer networks, and public switched telephone networks. For example in a computer network nodes are the computers which are connected by several transmission facilities for resource sharing and Internet can be defined as a globally available network of interconnected computer networks ([36]).

No matter what the scope of its use is, the motivation of network management, which includes design and planning efforts, is to set up a reliable communication channel between all participants to meet their needs. The participants are the customer sites that want to exchange traffic and the service provider who operates the network. In the field of telecommunication networks, the contract between the two parties includes a Quality-of-Service (QoS) agreement that guarantees a negotiated level of service quality. Hence, the impact of the performance of a specific design can be evaluated from both the customer's and the operator's perspectives. However, the operator's problem has always been more interesting for the operations research society.

Network management based on customer requests is the operator's responsibility. It is about planning, furnishing, coordinating, and monitoring network resources so as to maintain secure and reliable connection between customer sites at an acceptable performance level. In other words, where to locate how much capacity is the major decision the operator has to make. Furthermore, choosing the set of links and hence the routes to be used for each commodity is another critical component of the management process. Accordingly, significant research effort has been devoted to handle these problems since they are costly operations for the service providers.

Effective network management helps both the customer and the operator to benefit from cooperation. However, it requires close collaboration between the two parties. Customer's responsibility is to provide the information on predicted traffic flows, i.e. the traffic matrix, between its sites. On the other hand, operator is in charge of the network based on the input from the customer. Thence, the most critical entry for the decision process is the demand information, which can be provided in different levels of detail. The traditional approach in the network

design field is to use average estimates of the traffic demands, which are expected to represent the usual behavior of the system. But, as the way companies conduct their businesses changes, this simple approach is imminent to yield incompetent communication services. Firstly, inception of Internet has relaxed the constraints on collaboration and commercial relationships due to geographic locations. This has led to an increase in the number of end users of the corporate networks. Moreover, the variety of businesses has also increased. Besides, the ad hoc relations, which are common especially for the project-based businesses, call for flexible network structures. As a result, it is not likely to have a single estimate enclosing such uncertainties in the operational level without using overmuch resources. In this thesis, we concentrate on this problem and provide models that are robust to changes in traffic demand estimates for three quite important and well-known network design problems, namely the Virtual Private Network design, Open Shortest Path First routing, and the Network Loading Problem.

Virtual Private Networks (VPNs) are communications networks, which are constructed on top of other publicly available networks. They are private since they are devoted exclusively to their owners' needs. Moreover, they are virtual because rather than using dedicated connection tools like leased lines, they use 'virtual' routes over the Internet or the service providers' private networks. Their popularity is increasing since they provide the same level of communication security with devoted private networks but at lower cost. They also offer network flexibility especially when the underlying network is Internet. In that case, any VPN site on earth can be connected to the company network easily at low costs.

Improved network availability and secure data transfer at high quality are two major competitive advantages for VPN service providers. Since customers do not have to pay for their private networks, it is less costly to shift from one VPN provider to the other due to unsatisfactory levels of service, e.g., quite critical data transfers might be blocked frequently due to the insufficient capacity reservation on a network link. Hence the decision of changing the service provider becomes less critical. Anyhow there are quite a few companies offering a similar service. As a result, VPN service companies have started to appreciate the importance of efficient resource management in VPNs so as to ensure high service levels as

economically as possible.

The use of a rather general definition of feasible demand realizations in the VPN literature has been initiated with the inspiring paper of Duffield et al. [22]. They introduce the so-called *hose model*, where each terminal of the VPN is assigned a capacity for the total incoming and outgoing traffic it is expected to process. This was more a technical work discussing the mechanics and advantages of hose implementation. Subsequently, the operations research community has taken an interest in the problem and study several versions of it with respect to the hose definition and the routing restrictions. Although one of these types is relatively simpler, the increased flexibility in demand pattern definitions is paid in terms of higher modeling and solution efforts, generally. This is because it requires using a semi-infinite mathematical model. Thus, the related works are mostly centered around two approaches. The first one is investigating small constant factor approximations ([30], [31], [64]). The second method is dealing with the semi-infinite nature of the problem using iterative methods, which consider the valid demand realizations corresponding to the vertices of the demand polyhedron one by one ([12], [23]). More recently, Ben-Ameur and Kerivin [12] suggest using an arbitrary polyhedron to define feasible demand matrices.

Routing of demand requests between endpoints of a network is another critical component of network management efforts. This time, the concern is on using the available network resources in the most efficient way in order to lengthen the life of the network and avoid wastage of bandwidth. The routing technology and rules of data transmission are specified by routing protocols. Moreover, the admissible structure of paths on which to route traffic is also defined by some mechanisms. The most popular ones are Open Shortest Path First (OSPF) and Multi-Protocol Label Switching (MPLS).

In OSPF, each demand between an origin-destination pair is routed on the corresponding shortest path defined with respect to some link metric. The initial works on this problem use a fixed link metric. Later, following Fortz and Thorup [24], the idea of improving the OSPF performance by choosing the link weights so as to optimize a design criterion has become fairly common in the most

recent studies. On the other hand, MPLS is a more general routing mechanism since it restricts neither the structures nor the lengths of the paths. It is supposed to make a network easier to manage for improved customer service level.

Besides these protocols and mechanisms, the performance measure used to assess the goodness of a routing also affects the routing configuration in force. The most common criterion is the fair allocation of a given traffic demand among the network links, which is defined in terms of the utilization of the most congested link. It is desirable to evenly distribute the traffic load among all links in proportion to their capacities such that none of them becomes the bottleneck link due to over utilization. Accordingly, the overall objective is to keep the maximum link congestion at a minimum level. Similarly, when there is a set of feasible demands rather than a single one, the motivation is to choose the routing with the best worst case performance. This latter type of routing is called 'oblivious' since it is determined independent of a specific traffic matrix. Azar et al. [7] and Applegate and Cohen [3] incorporate demand uncertainty to determine the optimal oblivious MPLS routing. They consider any demand that can feasibly be routed on the network links without violating the link capacities. Later, again for MPLS routing, Belotti and Pınar [11] deal with two sets of demand definitions, which are formed by using the available information on the previous demand pattern observations for the network. The only study on OSPF routing belongs to Mulyana and Killat [53], where a rather restricted polyhedron is used.

The third problem we investigate in this thesis is the Network Loading Problem (NLP), which is about allocating transmission capacities on the links of the backbone network. The most significant difference between NLP and VPN design is that the capacity configuration must be integer this time. Several variations of the problem exist with respect to the numbers of facility alternatives and commodities. The most common problems are the one or two facility type problems with single or multiple commodities. The single commodity problem is very well studied since the extreme rays of its feasible set are fully characterized by Mirchandani [52]. On the other hand, the multi-commodity problem remains to be difficult. Magnanti et al. [50] extend their previous results for simpler versions of the problem to the two facility multi-commodity case and develop three families

of valid inequalities two of which are facet defining under very mild conditions.

To the best of our knowledge, there is no reference for NLP incorporating traffic uncertainty. This is because the problem is already quite difficult even for a deterministic demand matrix. Therefore, we can say that this field is totally untouched and needs further attention given that a design for a single traffic matrix assumption is likely to be incompetent for the dynamic nature of the modern business environment.

As mentioned above, current works on network design need to be improved by including a more general demand uncertainty definition. We believe in the necessity of designing networks, which are robust to fluctuations in demand estimates. It is important to enhance the practicality of our theoretical works by relaxing unrealistic assumptions. The contemporary business calls for higher flexibility and quality of service. We think that we contribute to the literature in that sense with our robust solutions to the three important network design problems.

As a result, we study the design of robust networks under polyhedral demand uncertainty, in this thesis. To form the basis for the later parts, we start with a comprehensive review of the literature in Chapter 2.

Then, we consider the VPN design problem in Chapter 3. As we have stated previously, despite the notable amount of available studies, there is no compact formulation for the problem under hose uncertainty. Moreover, all previous works concentrate on different types of the hose model. Although Ben-Ameur and Kerivin [12] made a first attempt to discuss general polyhedral demand uncertainty, they also make their numerical analysis using the hose model. We deal with these two deficiencies. We will first give a compact mixed integer programming model for the hose model. Then, we introduce the concept of robust VPN provisioning following the definition of Bertsimas and Sim [14, 15] and provide a novel, compact formulation for this new problem. Finally, we will implement a combined branch-and-price and cutting plane algorithm to solve the alternative path formulations with these two demand polyhedra.

In Chapter 4, we study the optimal oblivious OSPF routing problem where

we introduce demand uncertainty using an arbitrary polyhedron. We first show that the optimal oblivious MPLS routing under polyhedral demand definition can be determined in polynomial time. Then we provide a model for finding the corresponding OSPF routing with equal load sharing. Honestly, our model provides a quite flexible and practically defensible solution. This is because we use a general demand definition and equal load sharing in conformance with the available routing technology. Moreover, we let the service providers to determine the link metric so as to distribute traffic load fairly among the network links. Hence, our model yields a fair routing configuration viable under an infinite and uncountable number of demand scenarios. We also implement a branch-and-price algorithm strengthened by the inclusion of cuts to solve a formulation based on shortest path trees. We discuss some experimental results for the hose model and the robust approach of Bertsimas and Sim [14, 15].

We introduce polyhedral demands to the NLP literature in Chapter 5. To the best of our knowledge there is no other work on demand uncertainty in NLP. We start with an arbitrary demand polytope and give a compact mixed-integer programming model for this quite general problem. Then we project the formulation to a lower dimensional space, which reveals a very particular structure of our problem. Afterwards, we focus on the hose model and provide a thorough polyhedral analysis for this new problem. Following this analysis, we implement a branch-and-cut algorithm as an exact solution tool and present some numerical results.

Finally, we outline our contributions and discuss some future research directions in Chapter 6.

Chapter 2

Literature Review

A *private network* is the one whose owner acquires the exclusive use of its links, i.e., it is a network devoted solely to the use of its owner. On the other hand, a private network is *virtual* when it has some links across shared or public networks. Therefore, if a company owns a *Virtual Private Network (VPN)*, then the communication between its sites is partially or fully maintained over the public or shared network links. This disparity brings VPNs some competitive advantages, which will be discussed thoroughly in Section 2.2, over the devoted private networks. However, unless we specify otherwise, we will not discriminate between the two in this thesis since we deal with configuration oriented network management problems, which can be applied to either of the two cases.

In this chapter, we provide a review of the existing literature relevant to the problems we have studied. We will first discuss different approaches for estimating the most critical data component of network management efforts, that is the traffic matrix showing the demand flow requirements between some end points of the network, in Section 2.1. Then, we continue with VPN design literature in Section 2.2. Subsequently, Section 2.3 covers the existing research on traffic routing in IP networks. Finally, we outline the Network Loading Problem (NLP), which has been the focus of attention for a vast number of studies, in Section 2.4. We also provide a brief discussion of how this thesis would contribute to the available literature in Section 2.5.

2.1 Traffic prediction

Service providers need to know the amount of traffic client sites expect to exchange in order to determine the most efficient link capacity or routing configuration. The estimated amount of pairwise traffic flow between these sites make up the traffic matrix. Clearly, the amount and the accuracy of the information provided about the possible traffic matrices influence the final capacity allocation decisions as well as the functionality of the network. In this section, we will discuss different approaches, which are independent of the probabilistic nature of demand patterns, for traffic prediction. The methods reviewed here are certainly relevant to all chapters of the thesis.

The traditional way of defining the traffic matrix is the *pipe model*, where the client estimates the traffic flow between each pair of its end points and the service provider is supposed to provision the adequate bandwidth along the pipe of each path to ensure a negotiated level of service specified at the Service Level Agreements (SLAs) ([22]). Although any arbitrary routing pattern can be used, Jüttner et al. [43] mention that routing on the shortest path between each node pair gives the optimal solution with respect to the total capacity allocation on the network for the VPN under the pipe model.

The effectiveness of the resource allocation decision is very much dependent on how good the service provider and the clients cooperate, e.g., how accurate the information provided by the clients is or how effective the server is in resource management. The service provider experiences the major advantage of *implementing* a pipe model, namely the simplification of the resource management task. This is intuitive since the server is in charge of meeting a fixed traffic matrix estimated by the customer, which shapes the expected capacity requirement together with SLAs to some extent. However, this does not necessarily imply an advantage for both parties since the client is supposed to have somewhat precise information about the traffic matrix. This actually is not practically defensible given that most of the companies do not have such a database. Moreover, even if they have estimates for the average behavior of the network, it is almost sure that the demand realizations will be different than expectations. But since the

capacity is reserved for a specific link in the pipe model, it cannot be made available to another traffic request in case of an emergency. As a result, the likelihood of failing to carry some demand is not negligible. Let us give a small example to further clarify this issue.

Consider the following simple problem of deciding the optimal (i.e., resulting in the least total installation cost) number of devices of unit capacity to be installed on the links of the triangle-shaped network in Figure 2.1a to support the communication demands between the nodes. The number on each link gives the capacity installation cost of one unit capacity device. The communication demands are forecasted to be one unit of traffic flow among all pairs of nodes in both directions.

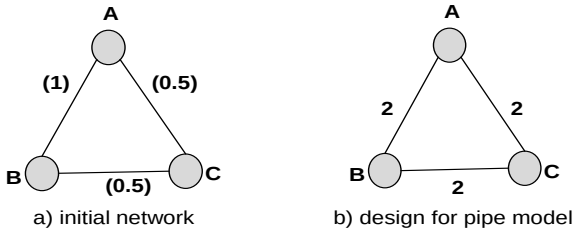

Figure 2.1: Example network for capacity loading.

Then the optimal capacity installation is given in Figure 2.1b with a total cost of 4. Now, suppose that the communication demand from node A to nodes B and C are realized to be 0.999 and 1.001, which are 0.1% less and more than the expected unit demand, respectively. It is immediately seen that for even such small deviations from the expected value, our optimal design in Figure 2.1b would be infeasible since link AB cannot support demand from A to C and C to A, simultaneously.

Finally, in the pipe model, each end point is expected to maintain a logical interface for each of its pipes, which increases the number of access links.

The second model for traffic matrix definition is the *hose model* of Duffield

et al. [22], which is firstly proposed within VPN design content. As the dependence of the contemporary business environment on IP technologies increases, the number of customer sites as well as the difficulty to estimate point-to-point demands also increase. It is not just the bigger networks but also the change in the communication pattern between the end points that make the resource allocation problem more complicated. For example, suppose that a VPN will be constructed over the Internet, which has actually become an important candidate for the physical network on top of which a VPN can be set up due to substantial improvements in the Internet technology. Note that a VPN site in almost anywhere on earth can be connected to the others over the Internet. This implies VPNs containing many more sites with a large variety of service requirements. Certainly, the traffic matrices are becoming more and more difficult to define. All these facts explain why the popularity of the hose model keeps increasing. As Italiano et al. [39] also mention, effective network management efforts should and must resort to approaches based on more flexible demand definitions like the well-known hose model.

Duffield, Goyal, and Greenberg, who have introduced the hose model in [22] initially, define their flexible model as follows:

> In the hose model, a VPN customer specifies a set of endpoints to be connected with common endpoint-to-endpoint performance guarantees. The connectivity of each point to the network is specified by a hose, comprising:
>
> - the capacity required for aggregate outgoing traffic from the endpoint into the network (to the other endpoints of the VPN)
> - the capacity required for aggregate incoming traffic out of the network to the endpoint (from the other endpoints of the VPN)
> - the performance guarantee for the hose, conditioned only on the *aggregate* traffic seen at the hose interface.

In brief, network design problem with the hose model can be defined as a problem in multi-commodity flow nature where the point-to-point demands are

not supposed to be known exactly. However, the validity of a traffic matrix can be judged based on its conformity to the ingress and egress capacities of all end points.

The fact that the hose model deals with a more flexible definition of the allowable traffic flows makes it more practically defensible. This is because the customers are not required to know the traffic distribution between the VPN sites, i.e., they do not have to give a specific detailed traffic matrix, but the aggregate traffic demand estimations for each site. Consequently, not only the variation in the individual estimates are reduced by aggregation but the clients can negotiate for more flexible Service Level Agreements (SLAs) as well. Moreover, suppose that we have a VPN site s. In the pipe model, the aggregate rate of traffic flow for s is the sum of the flow rates of the pipes associated with it. Duffield et al. [22] mention that the traffic rate at a hose can be smaller than the aggregate rate in the pipe model. Hence, another important strength of the hose model is that each VPN site needs to maintain a single interface to the provider's IP network. It is clear that all of the above advantages apply to the customers. The hose model is actually more difficult to handle for the service providers since the allowable range of traffic vectors is specified loosely. In this case, they only know that the total flows to and from each terminal can be anything less than some bandwidth values. So the traffic demand information is more uncertain and the service providers can reduce uncertainty by statistical multiplexing or resizing ([22]). The former approach implies aggregating all the traffic of a hose or the whole VPN together in order to stabilize the individual variations to some extent. On the other hand, by resizing, the provider can choose to be as risk averse as possible and allocate capacity considering the worst case demands. Alternatively, the capacity levels can be adjusted periodically based on the observed demand levels as well.

In the hose model, the underlying network over which the VPN will be defined is given as an undirected graph $G = (V, E)$ where V is the set of nodes and E is the set of edges. Moreover capacity reservation on network links has a cost and we are given a positive capacity reservation cost per unit for each edge $\{i, j\} \in E$, i.e., $c_{ij} \geq 0 \ \forall \{i, j\} \in E$. The customer asks for a VPN service where a set of

VPN terminals, i.e., $W \subseteq V$, are connected. Finally, for each VPN terminal $s \in W$, a couple of bounds on the total traffic entering and leaving s, b_s^+ and b_s^-, are given. Considering these bounds a valid traffic matrix d has the components d_{st} for the demand between each origin-destination pair (s,t) such that

$$\sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+ \quad \forall s \in W \tag{2.1}$$

$$\sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^- \quad \forall s \in W. \tag{2.2}$$

Based on this information we are required to find the minimum cost capacity reservation vector x such that each edge $\{i,j\} \in E$ equipped with capacity x_{ij} supports any valid traffic matrix defined as in (2.1) and (2.2).

The model in (2.1)-(2.2) is the most general definition for the hose model and it is also called as the *asymmetric case*. There are also two other hose characterizations, namely the *symmetric case* where $b_s^+ = b_s^- \ \forall s \in W$ and the *sum-symmetric case* where $\sum_{s \in W} b_s^+ = \sum_{s \in W} b_s^-$. Different approaches are proposed in the literature to handle the VPN design problem using these three hose models. The solution method and its complexity depends on the topology of the problem and the hose definition. We will provide a critical review of these approaches in Section 2.2. As a final comment, notice that the hose model can easily be used for the usual private networks by simply setting $W = V$ in (2.1) and (2.2).

The most recent traffic uncertainty model, which is the *polyhedral model*, is proposed in Ben-Ameur and Kerivin [12], where the feasible demand realizations are defined by an arbitrary polyhedron. In other words, a traffic matrix $d = [d_{st}]_{(s,t) \in W \times W : s \neq t}$ is valid if it belongs to a nonempty polytope D, which is defined by some customer specific inequalities. Based on this definition, it is clear that the hose model is also a polyhedral model where D is defined by (2.1), (2.2), and $\{d_{st} \geq 0 \ \ \forall (s,t) \in W \times W : s \neq t\}$. This approach is novel since it enables using additional linear inequalities together with the hose model in order to make D more representative for each customer. But now, the challenging task is to define D such that it is large enough to ensure some traffic flexibility but not so large

in order to avoid high reservation costs and wasted network resources. Some of the linear constraints Ben-Ameur and Kerivin offer to add to the hose model are as follows:

1. The traffic flow between some VPN terminals can be upper bounded by some value d_{max} using

$$d_{st} \leq d_{max} \quad \forall s,t \in S : t \neq s$$

where $S \subseteq W$.

2. If the terminal pairs are located in different countries it could be useful to group the terminals in each country together and then limit the total traffic flow between these groups using

$$\sum_{s \in A} \sum_{t \in B} d_{st} \leq (AB)_{max}$$
$$\sum_{s \in A} \sum_{t \in B} d_{ts} \leq (BA)_{max}.$$

This can be an example for the case where the connection cost between A and B is high, so the intergroup flows should be restricted to some extent.

3. Supposing that the routing cost is a function of the distance and traffic flow between the VPN terminals s and t, some upper bounds on flow can be determined based on the cost of online measurements. For example, Ben-Ameur and Kerivin propose to add the constraint

$$\sum_{s \in W} \sum_{t \in W \setminus \{s\}} \Theta(l_{st}) d_{st} \leq (1+\epsilon) \max_{d = \hat{d}_1, \ldots, \hat{d}_l} C(d)$$

where ϵ is a positive constant, $\hat{d}_i$ is the current vector of traffic demands at time i, $C(d)$ is the routing cost of a traffic demand vector d, which is defined as

$$C(d) \approx \sum_{s \in W} \sum_{t \in W \setminus \{s\}} \Theta(l_{st}) d_{st} \tag{2.3}$$

and $\Theta(.)$ is a function of the distance l_{st} between two VPN terminals s and t.

4. There might be some special nodes in the network such that the traffic flowing from terminal s to terminal t must visit. Then, constraints imposing such requirements as well as the ones specifying some characteristics of such special nodes can be added to the definition of D.

It is possible to find quite a few examples of such inequalities. They are all case dependent and helpful for making the hose polytope more representative. This is important in terms of reducing the uncertainty the provider is exposed to as a result of increased flexibility in the definition of the set of valid traffic matrices. Finally, note that although Ben-Ameur and Kerivin [12] discuss a polytope obtained by imposing additional constraints on top of a hose model, it is also possible to work with a totally new demand polyhedron without the bandwidth constraints. To sum up, Table 2.1 gives a comparison of the three different models for traffic demands definition.

Table 2.1: Comparison of the three models for traffic definition.

Criteria	Pipe Model	Hose Model	Polyhedral Model
resource sharing flexibility	Low	High	High
uncertainty to provider	Low	High	Medium
multiplexing gains	Low	High	High
ease of specification	Low	High	High
provisioning simplicity	High	Low	Medium

The common property of these three models is that they make no assumption on the probabilistic nature of the point-to-point demands. Since we will use the polyhedral demand definition throughout this thesis, we believe it is sufficient to mention the existence of those approaches using probability related measures for characterizing the set of feasible demand matrix scenarios and do not go into details on that type of traffic prediction.

2.2 Virtual Private Networks

As a result of globalization, the number of companies having their branches spread over a wide area is getting larger. Moreover, highly competitive business environment requires companies to respond quickly to fluctuations in their focus markets. Hence, the fast and secure communication within the company as well as between the company and its partners and customers becomes even more crucial. This is one of the reasons why the problem of network design is so popular. There are different approaches in the literature for designing communication networks ensuring the required levels of security at reasonable costs. One of the most widely encountered approaches is to use Virtual Private Networks (VPNs).

Christopher McDonald [51] defines VPNs as

> ... IP-based networks (usually the public Internet) that use encryption and tunneling to achieve one or more of the following goals:
>
> - connect users securely to their own corporate network (remote access),
> - link branch offices to an enterprise network (intranet),
> - extend organizations' existing computing infrastructure to include partners, suppliers, and customers.

This definition clearly states some advantages VPNs provide. First of all, the key feature of VPNs is that they use public networks, such as the Internet and public telephone lines. This means that the companies do not have to pay for expensive, private leased lines but transfer data over the public infrastructure and hence pay less. At this point one might ask 'How secure is the connection when the communication network is so much exposed to the public?'. The answer to this question mentions another strength of the VPNs. VPNs provide the same security and encryption features as a private network. Moreover, the use of public networks reduces the operational cost due to economies of scale while ensuring a wider area accessibility. The advantages of using VPNs can be summarized as follows:

1. VPNs take service from a public service, which has a large number of customers. Hence, the operational cost per customer would be less. Moreover the VPN clients do not have to call remote clients but the closest access point to the server, which reduces the cost of connection as well.

2. VPNs are quite flexible to accommodate a growth in the number of terminals since they use a network backbone available to the public. The addition of new client terminals is not difficult, which is appealing especially for the small- and medium-sized companies as they grow rapidly.

3. Since VPNs require a relatively short dedicated connection, the total cost of leasing expensive lines for communicating with the remote branches is reduced significantly.

4. Unlike the private networks the company can continue to use the existing IP infrastructure and equipment to connect the remote branches and customers.

Therefore, a VPN is an alternative to a private network. As such it should ensure at least the same quality of service level to be able to compete with the private networks. Hence the clients would naturally expect a VPN to be secure, highly available, and have predictable performance. By security we mean

1. *Confidentiality:* The privacy of the information being exchanged by the communicating parties should be ensured.

2. *Integrity:* It must be guaranteed that the information is not manipulated during the transfer over the public Internet.

3. *Authentication:* Only the permitted parties should be able to access the information.

The performance of a VPN is significantly affected by the Internet itself as well as the software and hardware applications. Another important factor is the level

of encryption of data. A higher level of encryption implies a longer transmission time ([33]).

Similarly, companies are very much dependent on the Internet Service Providers (ISPs) in terms of availability. It can be said that having insufficient capacity on the network links is very likely and some providers offer managed VPN services, where the VPN devices are owned and managed by the service provider so that the clients do not need to acquire capital equipment upfront, to overcome such problems to some extent.

2.2.1 An example use of a VPN

An example would be useful to further understand how a VPN works. Suppose that there is an employee who is traveling or who works at a branch in Istanbul and wants to communicate with the other branch in Ankara. Then, the steps followed to fulfill his request can be summarized as follows:

- *Step 0* The employee dials the local ISP and logs into the ISP network.
- *Step 1* To connect to the corporate network, the user sends a tunnel request to the destination security server on the corporate network. If the employee has the permission to access the corporate network, the server authenticates him and accepts the tunnel request.
- *Step 2* The data that the employee wants to send are encrypted by the VPN software and then they are sent through the tunnel over the ISP connection.
- *Step 3* The security server, which is at the other end of the tunnel, decrypts the received data and then forwards these decrypted data packets onto the corporate network.

As it can be understood from the above discussions, the VPN service provider should offer its clients a secure connection service, which is available very large portion of the time. Hence, there are quite a few issues that the service providers

can focus on to improve their service offerings. Since VPNs can be considered as alternatives for the private networks they should be competitive by all means. One of the most important component is security and much work has been done on it. Hence, due to the significant improvements in the IP security technologies different tunneling protocols like IPSec, PPTP, L2TP, and Socks5 are offered by the service providers. As it is mentioned by Duffield et al. [22], less attention has been paid to the network capacity reservation issue, which is important in order to maintain network availability. Noting that VPNs are mostly desired to support some functions, which are critical for the client, effective resource management in VPNs becomes even more crucial. As a result, we focus on how to determine the most effective capacity allocation so as to ensure network availability in Chapter 3 but first discuss the relevant literature in Section 2.2.2.

2.2.2 Resource management in VPN

The motivation of resource management for a VPN is to reserve sufficient capacity on the backbone network links for the VPN such that its point-to-point communication requests can be met by avoiding outages to the largest extent while keeping the total design cost at its lowest possible level. Obviously, the solution of the problem as well as its complexity is mainly affected by

1. the predicted amount of traffic between the customer sites, and
2. the topology of the solution, which is composed of the links with positive capacity reservation.

We have already covered the first issue on traffic prediction in Section 2.1, thoroughly. Besides, following the prominent work of Duffield et al. [22], hose model for traffic definition has become one of the most popular components of the VPN design efforts. Contributors including but not restricted to [12], [23], [30], [39], [43], and [45], have all used hose model due to its advantages over the traditional pipe model as mentioned in Section 2.1. Since we concentrate on uncertain rather than deterministic demands in this thesis, we will discuss available results

for VPN design under the hose model. So, in the rest of this section, we will review the current literature on four main types of the capacity allocation problem for VPNs under hose traffic definition.

In the solution of the resource management problem for a VPN, we have a collection of the backbone links on which positive capacity is reserved to maintain the transfer of traffic between the sites of that VPN client. The set of edges to be utilized varies for the same client depending on which one of the following three routing strategies is in action:

1. *Tree routing:* In this strategy, the union of the edges with positive capacity reservation must be a tree. In other words, each VPN site s is allowed to route the traffic to another site t along a unique path P_{st} such that the union of these paths, i.e., $\bigcup_{(s,t)\in W\times W:s\neq t} P_{st}$, is a tree. Kumar et al. [45] mention the following benefits of tree routing:

   - *Sharing of bandwidth reservation:* The capacity reserved on any link $e \in E$ can be shared by all VPN pairs on two opposite shores of the tree connected by e since link capacities are determined considering the aggregate requests and are not devoted any pair, particularly.
   - *Scalability:* For the case where the number of VPN sites is large, the number of paths might increase exponentially in a multipath routing scenario. Thus, tree routing scales better than its multipath counterpart since it contains a unique path between each VPN terminal pair.
   - *Simplicity of routing:* The application of the current routing technologies is less demanding.
   - *Ease of restoration:* In case of a failure the service provider does not have to restore each path individually. The subtree traversing a failed link can be restored as a group.

2. *Splittable routing:* It is also known as *bifurcated routing* where each site s is allowed to route the traffic to another site t along multiple paths, i.e., the set of paths P_{st}^1,..., $P_{st}^{l_{st}}$ with $l_{st} \geq 1$ can be used between each (s,t)

terminal pair. So, the solution topology, i.e., $\bigcup_{(s,t)\in W\times W:s\neq t}\bigcup_{i=1}^{l_{st}} P_{st}^{i}$, can be an arbitrary graph. Note that in this case not only the edges with positive transmission capacity but also the proportion of the traffic from s to t routed along each path P_{st}^{i} $\forall i = 1,..,l_{st}$ should be determined where the split factors and the set of paths might or might not depend on the traffic matrix , i.e., the routing might be dynamic or static[1].

3. *Unsplittable routing:* Finally another topology type, which is also discussed in the literature, is the one where each terminal s is required to route its traffic to site t along a unique path although the union of these paths is allowed to be an arbitrary graph. Note that, by definition, having each terminal site to route its traffic on a tree does not necessarily imply the final reservation graph to be a tree. This is especially mentioned here since Gupta et al. [30] state quite interesting results for this type of routing and it will be discussed in the next section together with other routing configurations.

Consequently, the VPN design problem under hose demand uncertainty is grouped into four main types, namely Sym-T, Asym-T, Sym-G, and Asym-G where Sym/Asym denotes the symmetric/asymmetric hose model and T/G represents the tree/splittable routing, i.e., Sym-T is the symmetric case with tree routing, Asym-G is asymmetric case with splittable routing and so on. There are different approaches for solving these problems in the literature and we review each problem one by one in the remaining parts of this section. However, we will first introduce some notation.

We consider an undirected graph $G = (V, E)$ where V is the set of vertices and E is the set of edges. Moreover, $W \subseteq V$ denotes the set of VPN terminals that we want to connect. The traffic matrix $d = [d_{st}]_{(s,t)\in Q}$ is a vector of nonnegative traffic demand between the VPN terminal pairs in the set $Q = \{(s,t) \in W \times W : s \neq t\}$[2]. The capacity reservation vector for $G = (V, E)$ is $x{:}E \rightarrow \Re^{+}$ and the support of x, i.e., the set of edges on which some capacity is reserved, is

[1] We consider static routing throughout the thesis .

[2] Although d is defined as a vector, the term *traffic matrix* is ubiquitous in the telecommunications literature, and we shall use the term *matrix* throughout to refer to vector d.

$I(x) = \{e \in E : x_e > 0\}$. Moreover each edge $e \in E$ is also associated with a unit capacity reservation cost $c_e > 0$.

2.2.3 Known results

2.2.3.1 Sym-T problem

Sym-T is the simplest type of the resource management problem, where bandwidth values are symmetric and the support of the reservation vector x, i.e., *I(x)*, is required to be a tree. The solution instance is a tree $T = \left(V^{'}, E^{'}\right)$ such that $W \subseteq V^{'} \subseteq V$ and each $e \in E^{'}$ is assigned some positive capacity. Moreover, since each edge has a positive unit reservation cost c_e, we can say that the cost of the instance T is $C(T) = \sum_{e \in E'} c_e x_e$. Then the purpose is to find the minimum cost tree T^*, which supports all valid demand realizations d in $D = \{d \in \mathbb{R}^{|Q|} : \sum_{t \in W \setminus \{s\}} d_{st} \leq b_s \; \forall s \in W\}$.

Gupta et al. [30] show that this problem is optimally solvable using a polynomial time algorithm based on shortest path computations. Their algorithm is based on an important property of the *tree routing* strategy, which ensures that each edge $e \in E$ should have a capacity of $\min\{b(L_e), b(R_e)\}$ where L_e and R_e are the two disconnected subtrees of a solution instance T obtained by deleting link $e \in E^{'}$ with $b(L_e) = \sum_{s \in L_e \cap W} b_s$ and $b(R_e) = \sum_{s \in R_e \cap W} b_s$. The property directly follows by the fact that one side would send as much as it can whereas the other side would receive as much as it can in the worst case and the link capacities should be sufficient to support such worst cases. Using this assertion, we can readily expect leaves of the tree T to be the VPN terminals. This is because $b_v = b_v^- = b_v^+ = 0$ for each nonterminal node $v \in V \setminus W$ and the capacity reservation on any edge e incident to v would have $x_e = 0$ according to the above discussions. Then the polynomial algorithm of Gupta et al. [30] uses these properties and some shortest path derivations to solve the Sym-T problem. Moreover, this method is also used to get a 2-approximation to the more general Sym-G problem ([30]).

2.2.3.2 Asym-T problem

In an Asym-T problem, the routing strategy is the same as the one in Sym-T, i.e., $I(x)$ is required to be a tree. But now we have some $s \in W$ such that $b_s^- \neq b_s^+$. This slight difference has a substantial effect on the computational complexity of the problem as [30], [40], and [45] independently show that the Asym-T problem is NP-hard.

Initially, Gupta et al. [30] show that the problem Asym-T is strongly NP-hard. This is because the case where the egress bandwidths are significantly larger than the ingress bandwidths can be reduced to a Steiner tree problem, which is known to be strongly NP-hard. In this case, ingress bandwidths are the determining factor for the capacity reservation on each edge. Actually each edge $e \in E$ is assigned the same amount of capacity, i.e., $\sum_{l \in W} b_l^-$. Therefore, the total cost depends on the number of edges in the solution, which means that the Steiner tree connecting the VPN terminals is the optimal solution with the least cost. So we cannot hope to find the exact solution in polynomial time. As a result, some algorithms are constructed to find constant factor approximations to the problem Asym-T.

The first approximation algorithm is initially proposed in [30] and then discussed in [45], as well. It is based on the similarity between the Asym-T and the Connected Facility Location (CFL) problems. In CFL, the motivation is to pick an optimal set $F \subset V$ of facilities such that each demand site i is assigned to an open facility where the facilities in F are connected by a Steiner tree and the total cost of facility opening, client assignment, and facility connection is minimum. Gupta et al. [30] and Kumar et al. [45] show that they can get a 10 factor approximation[3]. Furthermore, Kumar et al. [45] also propose a primal-dual approximation method as well as an adaptation of the polynomial time algorithm for Sym-T to the Asym-T problem. Finally, Swamy and Kumar [64] give the best approximation for the problem with a factor of 9.002.

[3]They guarantee that their algorithms find solutions, which are at most 10 times worse than the optimal solution of the corresponding Asym-T problem.

Later, Italiano et al. [40] worked on the Asym-T problem. Their focus is more on the fast recovery of a VPN network from a failure. The idea is to determine some *backup* links in advance and reserve some capacity on them in the least costly way. Then these links can be used in case of a failure in the primary links so as to ensure the functionality of the network. They argue that their problem is a variant of the optimal graph augmentation problem, which is NP-complete and thus present a 16 factor polynomial approximation algorithm.

2.2.3.3 Asym-G problem

Asym-G problem is the most general one since it allows the ingress and egress bandwidths for each terminal $s \in W$ to take arbitrary values and the solution can be an arbitrary subgraph of the underlying graph $G = (V, E)$. Due to its generality, it is widely discussed in the literature. We will discuss two similar solution approaches of Ben-Ameur and Kerivin [12] and Erlebach and Rüegg [23] for the Asym-G case in this section.

The VPN operator has to decide the fraction of each pairwise demand routed on each link for the Asym-G scheme. Moreover, link capacities should be determined considering all valid traffic realizations in the polytope D, which means an infinite number of possibilities. Two previous works [12, 23] deal with the semi-infinite nature of the problem by dynamically generating a set of traffic matrices corresponding to vertices of the demand polyhedron.

Iteratively, they solve a restricted problem defined for a valid traffic matrix d, which yields for each undirected link $e \in E$ values of the flow ($\tilde{f}_e^{st}$) of each pairwise demand d_{st} in both directions on e as well as the amount of capacity reservation ($\tilde{x}_e$) for e. Since the capacity reservation is made considering only one vertex of the demand polyhedron, it may not be feasible for all valid traffic matrices. So, a new vertex (traffic matrix) is sought by solving another linear program, where the $\tilde{f}$'s and $\tilde{x}$'s are considered as coefficients and the edge overload $\sum_{(s,t)\in Q} d_{st}(\tilde{f}_e^{st}) - \tilde{x}_e$ is maximized over this polyhedron D. If there exists a valid traffic matrix $[\check{d}_{st}]_{(s,t)\in Q}$ that is not supported by the capacity $\tilde{x}_e$ reserved on edge

e, i.e., if such overload is positive, a new link capacity constraint for the demand matrix $\check{d}$ with point-to-point demands $\check{d}_{st}$ is added to the formulation. If no such traffic matrix is found for any edge e, the current capacity reservations support all valid traffic matrices. Thus, these row-generation algorithms repeatedly improve a dual bound until primal feasibility is achieved. Although the focus is on critical demand values, $|E|$ linear programs must be solved at each iteration. In the special case of the hose model, these linear programs reduce to min-cost-flow problems, which yield an integral maximum flow on each edge ([12]). In [12] computational results are reported for a special class of instances with unbounded entering traffic in each terminal. In [23] the authors consider multi-path routing and check the validity of a reservation vector over a demand polyhedron with integral vertices. Finally, Gupta et al. [31] give a 5.55 approximation for the Asym-G problem, where the bandwidth values are assumed to be integer valued. This is the best bound known so far. However, we should mention that it is still an open question whether the Asym-G problem is actually NP-hard.

2.2.3.4 Sym-G problem

Sym-G problem is restricted in the sense that the ingress and egress bandwidth values are equal to each other for each terminal site. On the other hand, it is general since the routing patterns are allowed to be arbitrary graphs. Moreover, Sym-G can be considered as a special case of Asym-G and hence most of the time it is not discussed separately. The algorithms proposed for the Asym-G case can also be used for Sym-G as well. Therefore, rather than novel algorithms some important results and approximations are provided for the Sym-G problem. Namely, Gupta et al. [30] show that the optimal solution to the Sym-T for an instance is no worse than two times the optimal solution to Sym-G. But we should again mention that the NP-hardness of Sym-G is still not known for sure.

2.3 Traffic routing

Data transmission across the Internet networks is managed in conformance with the family of protocols *TCP/IP*. Internet Protocol (IP) uses a technology where the data sent over an IP-based network are divided into packets each of which contains the destination address. This technology is also known as the *packet-switched network* technique, which provides an efficient use of network resources and a lower risk of technical problems. On the other hand, the Transmission Control Protocol (TCP) regulates the transfer of data packets according to the bandwidth availabilities and the connection speed. It provides hosts an environment to manage themselves to some extent since it enables routers to update the set of available paths to destination as the network topology changes. In addition to the unconstrained routing schemes, a rule for path selection can also be applied so as to improve the routing performance. The most common criterion to assess the goodness of a routing is the link utilizations. The motivation is to minimize the maximum edge congestion so as to lengthen the life of the network and make the network more robust to link failures. All such efforts are called *traffic engineering*, which requires reacting to changes in the network conditions in order to ensure a higher customer service level and more efficient use of available resources.

Traffic engineering necessitates some performance objectives to give shape to the choice of paths. Moreover, routers should be equipped with necessary tools to abide what is tailored by the specific engineering efforts. Gateway protocols define the nature of traffic routing within and between autonomous systems. An *autonomous system (AS)* is a collection of IP networks and routers under the control of a unified entity that adheres to a common and well-defined routing policy ([36]). Traffic engineering for each single AS independently has been the center of a significant amount of work. In consequence, *Interior Gateway Protocols (IGPs)* for routing within ASs keep gaining more and more importance. There are two types of IGPs, namely the distance-vector routing protocols and the link-state routing protocols. The basic difference between the two types is the amount of information shared between the nodes of the network and the latter one is more

suitable for large networks.

The most common types of the link-state routing protocols are Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS). Both of these protocols use Dijkstra's shortest path algorithm with fixed link weights to determine the best routes to use and hence are conceptually similar. However, there are some technical differences between the two, which make OSPF more popular for IP networks.

Open Shortest Path First (OSPF) is a *link-state* routing protocol developed for IP networks in which routers send information to each other about the state of their adjacent links. Routers send the traffic between all nodes in an Internet network along the corresponding shortest paths composed of available links. These shortest paths are determined based on a metric established prior to network operations. There are different approaches for determining these metrics. The traditional approach is to fix link weights in advance based on some criteria like physical distances or the inverse of link capacities (Giovanni et al. [27]). However, fixing weights in advance leaves little space for performance improvement via traffic engineering in OSPF. Firstly, the set of admissible paths is limited since it is defined by a fixed metric. Moreover, such a restrictive approach is not forceful enough to pursuit any performance objectives. As a result, the concerns on increasing the flexibility of OSPF so as to incorporate traffic engineering has led to an interest in the management of link weights so as to optimize a design and routing criterion. We can cite Fortz and Thorup [24, 25], Holmberg and Yuan [32], Lin and Wang [47], Parmar et al. [56], Pióro et al. [57], Tomaszewski et al. [65], and Wang et al. [68] as examples for the most recent references for making weight management a component of traffic engineering efforts in OSPF routing.

Minimizing the maximum link congestion has been the major and most general performance measure in traffic engineering with the motivation of dealing 'fairly' with all network links. The 'fairness' of a routing can be measured by the utilization (i.e., the fraction of capacity used by data flow) of the most congested link. If some flow is distributed among the links in proportion to their capacities

such that none of them becomes the bottleneck link, then this measure would be small and the routing is relatively fair. Lin and Wang [47] state the following advantages of using the minimum of maximum link congestion as the performance measure:

- The optimal routing remains optimal if traffic demands grow uniformly.
- The optimal value of the maximum congestion can be used to derive upper bounds on some other performance measures like end-to-end delay or call blocking probability.
- The optimal routing ensures the most efficient use of network resources and avoids unnecessary capacity expansion.
- The routing decision made with minimax link congestion measure performs well with respect to other performance criteria in several networks like the call blocking probability in circuit-switched networks or the packet delay in virtual-circuit based packet networks.

In addition, we should also mention that it leads to similar depreciation levels for all physical links, which lengthens the operational life of the network. Given all these advantages, the popularity of the fairness criterion in traffic engineering is definitely justified.

In practice, the network operator determines the link weights in order to implement OSPF routing. Then, each router keeps a list of next links on any shortest path to any destination. This information is used to forward the incoming flow at each router in accordance with the configuration of the router. Current OSPF technology supports two possible configuration alternatives: routing on single paths or on multiple 'best' paths with equal splitting of flow. The latter case is known as *Equal Cost Multi-Path (ECMP)* rule. It is appreciated due to the significant bandwidth savings it provides.

There has been a lot of research to accomplish effective traffic engineering for OSPF routing. Different routing strategies as well as various ways to manage

them have been proposed. However, given the difficulty of the problem, some simplifications had to be made. The most common one is the assumption of a given demand matrix. Then again there is agreement among researchers that weight management is crucial to improve the performance of OSPF routing, and hence weight metric is not supposed to be given. Unfortunately, it is not trivial to determine a metric consistent with the capabilities of today's traffic forwarding technology and thus various strategies for controlling the weight metric are proposed. Such a problem can be thought of as a particular *inverse shortest path problem* (Zhang et al [71], Burton et al. [20]).

Weight management under ECMP is NP-hard (Wang et al. [68], Pióro et al. [57], Fortz and Thorup[24]) and the current technology does not support arbitrary load sharing. In order to tackle this difficulty either single path routing assumption or a couple of alternative strategies like the management of next hop selection or edge-based traffic engineering have been used. We cite Bley and Koch [18], Lin and Wang [47], and Tomaszewski et al. [65] as the examples for unsplit routing while we refer to Fortz and Thorup [25], Parmar et al. [56], Sridharan et al. [63], and Wang et al. [67] for the latter case of splittable routing with ECMP. References Broström and Holmberg [19], Giovanni et al. [27], Parmar et al. [56], Pióro et al. [57], and Tomaszewski et al. [65] also present mixed integer modeling examples for incorporating the ECMP rule. In Bley and Koch [18], Broström and Holmberg [19], and Pióro et al. [57], two-stage algorithms are used where the authors initially find an optimal routing scheme. Then, in the second step, they look for a metric that is compatible with the paths found in the first step, namely a metric according to which these paths are shortest paths. The drawback of these approaches is that not all configurations are guaranteed to be realized as shortest paths. Although Wang et al. [68] show that a class of routes with some property can be converted to shortest-paths, still no complete description of admissible routing schemes is available. Alternatively, Fortz and Thorup [24], Giovanni et al. [27], Lin and Wang [47], Parmar et al. [56], and Wang et al. [68] prefer to consider the optimization of a design criterion and the link metric, simultaneously.

The efforts on traffic engineering in OSPF have almost always been centered

around finding a weight setting, which is the best for a single deterministic demand matrix and the given network topology. It was Fortz and Thorup [25] to investigate the problem of determining a single weight setting efficient for some anticipated changes in the network topology and demand matrices, firstly. They mention that a link failure might spoil the optimality of the current routing and hence a change in weight metric is needed. A similar situation is also valid in case of periodic demand changes. Unfortunately, frequent weight changes might be disruptive for the network since the routes must be recomputed each time. Thence, Fortz and Thorup [25] focus on designing a tool for weight tuning, which requires the minimum amount of changes to adopt to disruptions in the network due to link failures or a demand matrix realization from a given set of possibilities. In terms of the changes in demand matrices, they consider the transitions between periods with different demand patterns like the day time and evening. So, they try to find a single weight setting good for all periods like the whole day using the local search heuristic they have presented in [24].

As the concept of a 'set of demand matrices' rather than a single one looms large in network design problems, several attempts for traffic engineering with multiple demands started to come into being. This has led to a modification in the performance parameter such that each routing is assessed according to its worst case performance for any traffic realization in the set of feasible demands. Such a routing is called *oblivious* since the path between every node pair is chosen independent of the current demand matrix. So, the goal of oblivious routing is to find a set of fair routing paths for all source-sink pairs regardless of the demand matrix. This problem has been investigated by Azar et al. [7] and Applegate and Cohen [3], previously. In both papers, the set of feasible demands contains all traffic matrices, which can feasible be routed on the underlying network. Moreover, they both consider the unconstrained and general routing mechanism called *Multi-Protocol Label Switching (MPLS)*, which does not impose any restriction on the path lengths unlike OSPF or IS-IS. Azar et al. [7] provide an LP model, which has exponentially many constraints with a polynomial time separation oracle and a cutting plane algorithm. On the other hand, the LP model of Applegate and Cohen [3] is polynomial in size, where they make a simplifying assumption, which

is shown later to be invalid by Belotti and Pınar [11]. They assume that at least one of the network links would be used to full capacity in the worst case but Belotti and Pınar [11] give a counter example showing that this does not have to be the case[4]. More recently, Belotti and Pınar [11] consider box and ellipsoidal uncertainty representations for oblivious MPLS routing. They focus on the case where traffic demand is assumed to have some lower and upper bounds as well as the case where the mean-covariance information of random demand is available. On the other hand, Mulyana and Killat [53] investigate the oblivious OSPF routing problem where the set of feasible demand matrices is a polyhedral set defined by simple outbound traffic constraints. They use a simulated annealing based heuristic, which might yield a solution with link capacity violations. Furthermore, authors mention that a final validation is necessary for the approval of the optimal solutions, which is obviously a deficiency of their work.

2.4 Network Loading Problem

Network Loading Problem (NLP) is a special type of the well-known Capacitated Network Design problem. Although it is mainly associated with telecommunication network design, it can also be extended to different contexts like transportation planning, freight loading on trucks or computer networks as mentioned in Magnanti et al. [50] and Berger et al. [13].

For a given undirected network $G = (V, E)$, the traditional NLP deals with the design of a network by allocating (loading) discrete units of capacitated facilities on the links in E so as to support estimated pairwise demands between some endpoints in V. Similar to the VPN design problem we have discussed in Section 2.2, the purpose is again to determine the least costly design. However, the important distinction between the two is that the final capacity configuration must be integer for NLP.

For the telecommunication networks, various capacitated facilities can be used

[4]This will be explained further in Chapter 4.

for delivering data communications in the form of voice, data, graphics, video etc. Since 1990s digital facilities have been the most frequently used technology due to the deployment of digital switches. There is a small set of digital facility alternatives each with different service levels, i.e., data transmission rates. For example, DS0 (Digital Signal Level 0) corresponds to the basic level with a transmission rate of 64 kilobits per second (Kbps), which is the bandwidth used for one telephone voice channel, normally ([34]). Then the series of standard digital transmission rates, which is called as the *digital signal X series*, is defined with respect to DS0 such that DS1 is equivalent to 24 DS0, DS2 to 4 DS1, DS3 to 7 DS2, and DS4 to 6 DS3 in terms of their transmission rates. The choice for which of these service levels to lease depends on the customer's demand volume as well as communication quality requirements. For example DS0 is ideal when the transmission volume is expected to be low whereas DS3 should be deployed when the customer needs various forms of data to be transmitted at high speed and quality. A single type or multiple types of facilities can be used for each customer and the decision naturally depends on the cost of each service level. Obviously, management of the tariff structure is complicated and the economies of scale prevails, i.e., the cost of leasing 24 DS0 facilities is more than the cost of 1 DS1 facility ([50]). However, the cost management side of the problem is not in the scope of the traditional NLP, which deals with the design of a network given the cost of technology.

NLP under deterministic traffic demands is widely studied in the literature. The generic problem has several versions. Namely, the number of different facility types available for installation give rise to variants of NLP. The most frequently studied cases are the single-facility ([5], [16], [49], [52], [58], [66]) and two-facility ([17], [28], [50]) versions whereas Magnanti and Mirchandani [48] discuss the three-facility problem as well. In addition, different cost functions are used, and the most common distinction is due to the existence of flow costs. Actually, NLP is considered as a special case of the Capacitated Network Design problem where the flow costs are zero. We cite Avella et al. [6], Bienstock et al. [16], Magnanti and Mirchandani [48], Magnanti et al. [49, 50], and van Hoesel et al. [66] for the zero flow cost problems. On the other hand, Bienstock and Günlük [17],

Günlük [28], and Rardin and Wolsey [58] study the Capacitated Network Design problem where the cost of routing is included in the overall network cost. The routing of demands is another source of diversity. Although the multi-path routing (*bifurcated routing* or *splittable flow*) is more popular ([5], [6], [16], [17], [28], [48], [49], [50], [58]), some important studies with single path routing (*non-bifurcated routing* or *unsplittable flow*) are also available ([5], [13], [29], [66]). Nonetheless, the Capacity Expansion Problem (CEP), where the decision is to determine a capacity installation plan by expanding the current edge capacities so as to route multi-commodity demands, is also closely related with the network loading problem. We can cite Atamtürk and Günlük [4], Atamtürk and Rajan [5], Bienstock and Günlük [17], Berger et al. [13], and Günlük [28] for polyhedral results on the capacity expansion problem. The latter reference also provides a branch-and-cut algorithm for CEP.

Since NLP is strongly NP-hard, there have been various efforts for solving it as efficiently as possible through the use of alternative formulations, heuristics, and by a thorough polyhedral analysis. Magnanti and Mirchandani [48] work on the single commodity restriction of the network loading problem with up to three facility alternatives. They try to reformulate the problem as a shortest path problem and solve it using heuristics. They show that the single facility case can be solved as a shortest path problem, which is not true for the two- and three-facility versions in general. Although the single commodity is not a practical assumption, the authors argue that their results could be an edge for the multi-commodity problems. Alternatively, single arc/edge restrictions of the problem have also been the focus of attention. Magnanti et al. [49] provide the single arc design problem, which is solved efficiently by a greedy method. Moreover, van Hoesel et al. [66] concentrate on the polyhedral properties of such subproblems since they can be seen as a relaxation of the original problem and the corresponding polyhedra are closely related. Similarly, Atamtürk and Günlük [4] study the network design arc set with variable bounds, which they emphasize to be a common component of a good number of network design problems.

Even though there have been different approaches in the literature for handling the network loading problem efficiently, most of them investigate the polyhedral

properties of the related problems. The common approach is to define some strong valid inequalities to strengthen the linear programming relaxations and thus improve the performance of the branch-and-cut or branch-and-bound based solution methods. Moreover, projection of the feasible set on to the space of discrete design variables has been a common point of interest ([5], [6], [16], [17], [48], [49], [50], [52], [58]). On the other hand, Berger et al. [13] discuss a tabu search heuristic.

In all references on deterministic NLP, point-to-point communication requirements are assumed to be given, and the future demand forecasts are used. Obviously, the demand between each origin-destination pair can be considered as a single commodity, and hence the overall problem is of a multi-commodity flow nature. The problem for single commodity flow with two facility types is very well studied and the polyhedra of feasible flows is fully characterized ([52]). The same problem with multi-commodity flows remains very hard, and its solution requires the use of metric inequalities to define the projection of the corresponding polyhedron on the space of discrete design variables ([6], [16], [17], [28], [48], [49], [50], [52], [54], [58]). In brief, the current efforts for the multi-commodity problems are mostly centered around the metric or mixed integer rounding (MIR) inequalities.

All these references consider different versions of the Capacitated Network Design problem, which are closely related with each other. NLP is also a special type of the general problem where there is no variable flow cost. Hence, we will conclude this section with a brief discussion of the studies particularly on NLP. Initially, Magnanti and Mirchandani [48] consider the single commodity problem with up to three facility alternatives. They investigate whether or not their problem could be modeled and solved as shortest path problems. They show that the answer is affirmative for the single facility problem and for some special versions of the two- and three-facility problems, where the same economic break-even points are valid between all facility types on all network links. In addition they show that the shortest path solutions yield asymptotically good bounds for the general two- and three-facility problems, which are actually NP-hard.

In sequel, Magnanti et al. [49] study the polyhedral structures of two integer programming subproblems of the single facility NLP. The first problem is a single arc design problem obtained by decomposing the Langrangian relaxation of NLP for each arc. On the other hand, the second problem is NLP on a three node network where there is a traffic flow between all three nodes. Authors give a complete description of the convex hull of feasible solutions for both problems. The importance of this work is that it has been the inspiration of several extensions on network design problems. Later on, the same authors extended these results to the case of two facility NLP with multi-commodity flows in [50]. First they generalize the *cutset inequalities* of Magnanti and Mirchandani [48]. A cutset is the subset of the links of the network whose removal would split the network into two unconnected components. Then, the cutset inequalities ensure that the capacity of each cutset is enough to support the total estimated demand between its two shores. Authors show that cutset inequalities are facet defining for the convex hull of the feasible solutions of the problem under very mild conditions. Then, they prove that the generalization of the facet defining inequalities of the single arc design problems (*arc residual capacity inequalities*) leads to a family of facet defining inequalities for the two facility problem. Finally, they discuss how the *3-partition inequalities* in [49] can be adopted to the more general two facility problem. Although the latter inequalities are not facet defining, Magnanti et al. [50] offer to use them together with the previous two families so as to improve the linear programming relaxation of the original two facility problem.

Mirchandani [52] investigates the projections of the single commodity and multi-commodity versions of the two facility capacitated network loading problem. He projects out the flow variables and discuss the polyhedral properties of these equivalent problems in a lower dimensional space. His first contribution is to give a complete characterization of all facet defining inequalities for the single commodity case. Then, he uses these results to give some necessary conditions for the multi-commodity problem. The feasibility check of a given capacity configuration for the multi-commodity case is not an easy task and requires the use of metric inequalities. The difficulty is due to the fact that we may still fail to find a feasible multi-commodity flow even if the total capacity of every cutset is

more than the total demand across the cutset. The second contribution of Mirchandani is to strengthen the current results on multi-commodity flow problem by defining the necessary conditions for the extreme rays of the corresponding projection cone.

Most recently, Avella et al. [6] consider the single source single facility NLP. They study the projection of the problem onto the space of discrete design variables and describe a new class of *tight metric inequalities*, which completely describe the convex hull of integer feasible solutions of the problem.

Unlike the Uncapacitated Network Design problems, the linear programming relaxations of the Capacitated Network Design problems yield weak lower bounds. Hence, the motivation for all these polyhedral analyses for describing strong valid inequalities is to get stronger lower bounds. Thereof, investigating efficient separation algorithms for these inequalities is another important issue. Magnanti et al. [50] use heuristics to determine the violated cutset, residual capacity, and 3-partition inequalities for the two facility multi-commodity NLP problem. Then, Bienstock et al. [16] consider the single facility multi-commodity Capacitated Network Design problem and show that the separation of a special type of the rounded metric inequalities, namely the *partition inequalities*, can be done in polynomial time. Later, Atamtürk and Rajan [5] give a polynomial time separation algorithm for the residual capacity inequalities of Magnanti et al [49] with single facility multi-commodity problem with splittable routing. On the other hand, Atamtürk and Rajan [5] show that the separation problem for the unsplittable version of the same problem is NP-hard, which requires solving a knapsack problem. Consequently, van Hoesel et al. [66] discuss simple heuristics for three families of valid inequalities for the undirected edge capacity polytope. Finally, Avella et al. [6] give two-stage heuristics for separating the metric inequalities.

2.5 Discussion

The assumption of a known demand matrix might be acceptable to some extent for small networks with a few end users among whom the traffic demand patterns have matured. However, such networks are not common for the current business environment. In the simplest case, the demand patterns would mostly change during a day and the network operators should determine a design, which accommodates such changes in the least costly manner without using excessive capacity. Moreover, companies need larger networks with many end points. For such networks, it is almost impossible to predict the pattern or the type of data transmission requests. As a result, a design based on a single demand matrix estimate might be incompetent to ensure a competitive network performance. Thus, the necessity of the design of networks for a general demand definition is inevitable.

Among the three problems, the highest number of works using demand uncertainty belongs to the VPN design literature. The foregoing efforts focus on finding good approximation algorithms or solving the problem by considering the vertices of the demand polyhedron one at a time since the problem is of semi-infinite nature. So, there is no compact formulation available for the problem. Moreover, to the best of our knowledge, there is no implementation of the problem with demand polyhedra other than the hose model. We deal with these two deficiencies and present compact models for the hose uncertainty definition as well as a less conservative robust variant together with an exact solution method in Chapter 3.

On the other hand, there are very few works on traffic engineering under demand uncertainty most of which are concerned with unconstrained routing rather than OSPF. We are not aware of any effort other than the recent work of Mulyana and Killat [53] for OSPF routing. They try to handle a rather restricted definition of hose model using a simulated annealing based method. However, there is no compact formulation or exact solution tool available for the problem. So, we are confident to say that there is an important need for further research on this problem. Hence, as a first effort, we address traffic engineering in OSPF routing

under general demand uncertainty in Chapter 4 where we provide compact mathematical models for MPLS routing and OSPF routing under demand uncertainty together with an exact solution tool for the latter problem. Our results lead to a comparison of the two technologies in quite general environments based on which we discuss if OSPF routing could become comparable with the unconstrained MPLS routing.

Finally, we study the Network Loading Problem for which we are not aware of any previous work using demand uncertainty. This is mainly because the generic multi-commodity case with a deterministic demand matrix is already difficult and lacks satisfactory results for efficient solution methods. However, we introduce the general demand definition to the field and obtain a quite elegant property for the hose model. We discuss our model and the exact solution method in Chapter 5. As a result, we can now conclude the review of the existing results for these very important network design problems and continue with our contributions in the rest of the thesis.

Chapter 3

Virtual Private Network Design

3.1 Introduction

Virtual Private Networks (VPNs) are network services built over an existing public network to provide quality of service, flexibility, and security while saving costs through link multiplexing. The use of public networks reduces operational costs due to economies of scale while ensuring a wider area accessibility and communication security through encryption. Flexibility and cost effectiveness have turned VPN solutions into a billion dollar industry.

In this chapter we address a network design problem that arises when provisioning VPNs: Given a set of terminals that must communicate with one another and a set of possible traffic patterns, sufficient capacity has to be reserved on the links of the large underlying public network to support all valid traffic patterns while minimizing cost. As discussed in Section 2.2, the problem admits several variants depending on the desired topology of the reserved links and the nature of the traffic data uncertainty.

The rest of the chapter is organized as follows: In Section 3.2, we provide the notation and a formal definition of the problem. Section 3.3 includes the linear mixed-integer programming formulations we develop for several variants of the

VPN provisioning problem. Namely, we present a compact flow-based linear MIP formulation for the problem with the classical hose traffic model in Section 3.3.1, and for a new, less conservative, robust variant relying on the mostly available traffic statistics in Section 3.3.2. These flow-based formulations allow us to solve optimally medium-to-large instances with commercial MIP solvers. For both hose uncertainty and robust uncertainty models, we also discuss cutting planes for obtaining lower bounds. Finally, to close the discussion of the models, in Section 3.3.3 we give the path-based formulations more suitable for the exact solution method detailed in Section 3.4 where we describe a combined branch-and-price and cutting plane algorithm to tackle larger instances of the three problem versions. Computational results obtained for several classes of instances are reported and discussed in Section 3.5. Lastly, Section 3.6 contains some concluding remarks.

3.2 Problem definition

We will first introduce the notation of this chapter. The underlying network over which a VPN has to be provisioned is represented by an undirected graph $G = (V, E)$. Each edge e in E, also denoted by $\{h, k\}$ to emphasize the two end-nodes h, k in V, has a per-unit non-negative reservation cost c_{hk}. Let $W \subseteq V$ denote the set of terminals that need to communicate with one another. For each ordered pair of terminals (s, t), with $s, t \in W$ and $s \neq t$, the nonnegative d_{st} represents the amount of traffic that has to be routed from s to t. We denote by Q the set of all ordered pairs of distinct terminals, namely $Q = \{(s, t) \in W \times W : s \neq t\}$. In the hose model, two non-negative bounds b_s^+ and b_s^- are specified for each terminal s in W, and the traffic demands d_{st} are assumed to be non-negative values that belong to the uncertainty set U_{AsymG} defined by the following inequalities:

$$\sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+, \quad \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^- \qquad \forall s \in W \tag{3.1}$$

$$d_{st} \geq 0 \qquad \forall (s, t) \in Q. \tag{3.2}$$

Throughout this chapter we make the realistic assumption that traffic is *unsplittable*[1], i.e., for each demand pair $(s,t) \in Q$ the traffic demand d_{st} is routed along a *single path* from s to t. Provisioning a VPN then consists of reserving capacity on the edges and selecting a single routing path for each demand pair to support all valid traffic matrices while minimizing the total reservation cost. In the VPN provisioning problems considered in the present chapter no assumption is made on the integrality of demand values even if the bounds b_s^+ and b_s^- are integral. In other words, the resulting design should support integral as well as non-integral traffic demand matrices from the uncertainty polyhedron (3.1)–(3.2). Therefore, the network design problem of the present chapter, which has a multi-commodity flow structure, is both continuous and discrete in nature since fractional amounts of capacity can be reserved on the links while demands must be routed along single paths.

In addition to the asymmetric case, we also have the symmetric problems *Sym-G* and *Sym-T*, where a threshold b_s is given for the sum of incoming and outgoing traffic for all terminals $s \in W$. We call the set of traffic demands feasible for the symmetric version as U_{SymG}, which is defined by the following set of constraints:

$$\sum_{t \in W \setminus \{s\}} d_{st} + \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s \qquad \forall s \in W \tag{3.3}$$

$$d_{st} \geq 0 \qquad \forall (s,t) \in Q. \tag{3.4}$$

Figure 3.1 illustrates an instance of *Sym-G*, the corresponding optimal solution, and the routing of two traffic matrices within U_{SymG} with the related routing. The network topology is given in part (a) with edge capacity costs where the terminal nodes 3, 5, 6, and 7 are emphasized and the upper bounds b_s are shown. We depict the optimal solution in part (b) with edge capacities. Moreover, two sets of traffic demands within U_{SymG} are given in part (c) and part (d), with the related routing. For example, in part (c) the traffic demand is such that $d_{76} = 5$, $d_{67} = 3$, $d_{73} = 6$, and the remaining d_{st}'s are equal to zero.

[1]Although the case where each traffic demand can be arbitrarily split and routed along several paths is considered among others in [23], multipath routing is currently hardly implementable in VPNs because packets related to a single flow may arrive out of sequence and thus cause critical problems at the Transfer Control Protocol (TCP) level.

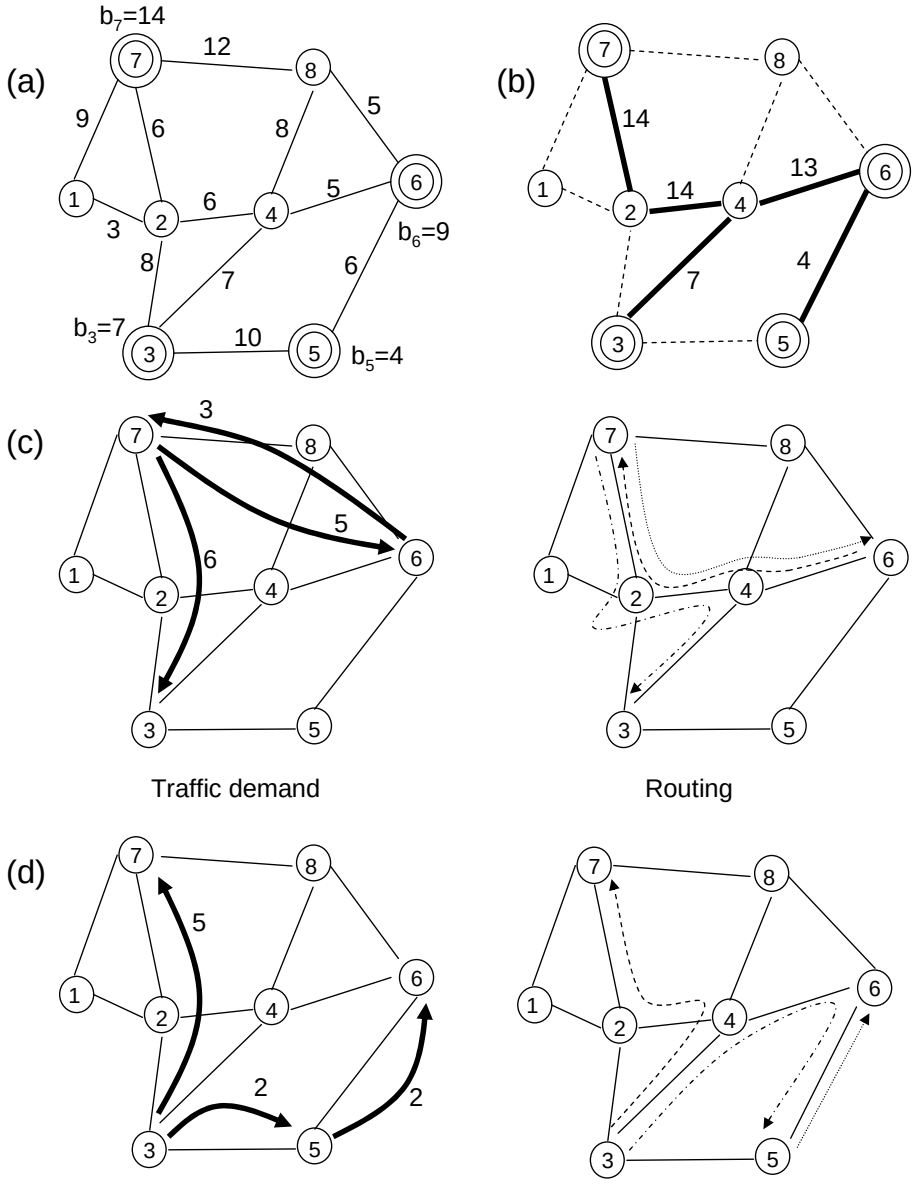

Figure 3.1: An instance of *Sym-G*.

Traffic uncertainty is a crucial feature in VPN provisioning. Since the hose model makes very weak assumptions (only imposes upper bounds on the inflow and outflow of each terminal), it may lead to excessive capacity reservation. This traffic model is a special case of the so-called *polyhedral model* in which the set of valid matrices is defined by an arbitrary polyhedron ([12]). In principle the polyhedral model allows us to focus on smaller subsets of traffic matrices than the hose model, but it is unclear how to actually define realistic polyhedra that would lead to less conservative VPN reservations.

From the application point of view, a service provider simultaneously provisions a number of VPNs for different customers over a certain time period and the service level agreements are re-negotiated on a regular basis. While the hose traffic model is adequate for new VPNs in the absence of precise traffic predictions, it is clearly overly conservative for VPNs that are already provisioned. Since service providers collect detailed terminal-to-terminal traffic statistics for each VPN, a traffic uncertainty model which exploits the available statistics is needed to avoid over-provisioning. Therefore, we will also investigate a less conservative robust variant of the problem, which makes use of the traffic statistics that are available for existing VPNs.

In practice, service providers face an incremental reservation problem: while provisioning a given set of VPNs, they receive a request for a new VPN or for changes to the set of terminals of an existing one. Since the additional capacity requirements for a single VPN are in general very small compared with the overall network capacity, it is reasonable from an application point of view to focus on the uncapacitated versions of the VPN provisioning problem ([21]). Due to the absence of capacity constraints and the fractional capacity that can be reserved on each link of the backbone network, the incremental problem can then be decomposed into a sequence of single VPN provisioning problems with appropriate traffic uncertainty models. Hence, we will concentrate on the uncapacitated versions of the problem under two different types of traffic uncertainty in the current chapter.

3.3 VPN provisioning models

In this section, we discuss our main contribution to the VPN provisioning problem. Firstly, we develop new, compact, linear mixed-integer programming (MIP) formulations for the *Asym-G* and *Sym-G* variants of the VPN provisioning problem under the hose model of uncertainty (cf. Proposition 3.3.1). Although the computational complexity status of *Asym-G* is still open, medium-to-large instances of our models turn out to be solvable by the off-the-shelf mixed-integer optimizer Cplex 8.1 within short computing time. Moreover, we introduce the concept of robust VPN provisioning following the definition of Bertsimas and Sim [14, 15]. In particular, we present a new, compact linear MIP formulation of the corresponding problem (cf. Proposition 3.3.2) and show that the robust VPN provisioning problem is NP-hard (cf. Proposition 3.3.3).

3.3.1 VPN provisioning problem with the hose traffic model

For new VPNs or existing ones in which at least a new terminal is added, demand statistics are not available or not reliable due to the lack of information about the traffic generated by the new terminals. The hose uncertainty model is then well suited for the demands involving at least a new terminal.

Consider the general variant *Asym-G* with asymmetric traffic matrices. The network cost, which depends on the capacity to be reserved so as to route all demands, has to be minimized. Since traffic is assumed to be *unsplittable*, each demand has to be routed along a single path from origin to destination.

3.3.1.1 From constant traffic matrices to the hose model

Let us first consider the capacity reservation problem aiming at minimizing cost while supporting a single traffic matrix $d = [d_{st}]_{(s,t)\in Q}$ and note that a straightforward minimum cost flow formulation can be solved in polynomial time. We

define two classes of variables. For each edge $\{h,k\}$, the continuous variable x_{hk} represents the capacity to be reserved on $\{h,k\}$. Let $A = \{(h,k) : \{h,k\} \in E\}$ denote the underlying set of oriented arcs. For each oriented arc $(h,k) \in A$ and oriented pair of terminals $(s,t) \in Q$, the binary variable f_{hk}^{st} corresponds to the flow on the oriented arc (h,k) for the terminal pair (s,t), i.e.,

$$f_{hk}^{st} = \begin{cases} 1 & \text{if arc } (h,k) \text{ is used to route demand } d_{st} \\ 0 & \text{otherwise.} \end{cases}$$

This leads to the flow formulation:

$$\min \sum_{\{h,k\}\in E} c_{hk} x_{hk} \tag{3.5}$$

$$\text{s.t.} \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V,\ (s,t) \in Q \tag{3.6}$$

$$\sum_{(s,t)\in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) \le x_{hk} \qquad \forall \{h,k\} \in E \tag{3.7}$$

$$f_{hk}^{st} \in \{0,1\} \qquad \forall (h,k) \in A,\ (s,t) \in Q \tag{3.8}$$

$$x_{hk} \ge 0 \qquad \forall \{h,k\} \in E, \tag{3.9}$$

where the objective function corresponds to the VPN reservation cost. Constraints (3.6) ensure demand satisfaction by imposing a unit flow between each oriented pair of terminals. For each edge $\{h,k\}$ the corresponding constraint (3.7) guarantees that the capacity x_{hk} reserved on $\{h,k\}$ is sufficient to carry the total traffic flowing through it.

If all the demands d_{st} are precisely known, (3.5)-(3.9) is a linear mixed-integer program. Due to the non-negativity of the costs c_{hk} and of all variables, constraints (3.7) are satisfied as equalities in any optimal solution to (3.5)-(3.9).

Therefore, the continuous variables x_{hk} can be omitted and we have the equivalent formulation (VPN_{nom}):

$$\begin{aligned}
&\min \sum_{\{h,k\}\in E} c_{hk} \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \\
\text{s.t.} \quad &\sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} && \forall h \in V,\ (s,t) \in Q \\
&f^{st}_{hk} \in \{0,1\} && \forall (i,j) \in A,\ (s,t) \in Q
\end{aligned}$$

with only binary variables. Since there are no capacity constraints and fractional capacities can be reserved on the edges, the constraint matrix is totally unimodular and the problem is solvable in polynomial time ([60]).

If the traffic matrix d is subject to the hose uncertainty model, a first formulation is obtained by requiring that capacity constraints (3.7) hold for all traffic matrices satisfying constraints (3.1)-(3.2), or respectively (3.3)-(3.4). Thus, we obtain the following semi-infinite MIP formulation ($HOSE_{sem}$):

$$\begin{aligned}
&\min \sum_{\{h,k\}\in E} c_{hk} x_{hk} \\
\text{s.t.} \quad &\sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} && \forall h \in V,\ (s,t) \in Q \\
&\sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \le x_{hk} && \forall d \in U_{\text{AsymG}}(U_{\text{SymG}}),\ \{h,k\} \in E \qquad (3.10) \\
&f^{st}_{hk} \in \{0,1\} && \forall (h,k) \in A,\ (s,t) \in Q \\
&x_{hk} \ge 0 && \forall \{h,k\} \in E.
\end{aligned}$$

We have discussed how this problem is solved using iterative algorithms in Ben-Ameur and Kerivin [12] and Erlebach and Rüegg [23] in Section 2.2.3.3. They had to use such algorithms so as to deal with the constraints (3.10), which are uncountable and semi-infinite in number. However, unlike these approaches where the traffic matrices corresponding to the vertices of the demand polyhedron

d are considered one by one, we propose to use a compact MIP problem in the next section.

3.3.1.2 A compact linear MIP formulation for the problem with hose model

Unlike in the above-mentioned references ([12], [23]), we simultaneously consider all demand constraints and derive a compact linear MIP formulation that avoids the semi-infinite MIP formulation.

Proposition 3.3.1. *The $HOSE_{sem}$ model for the* Asym-G *problem is equivalent to the following linear mixed-integer program:*

$$\min \sum_{\{h,k\}\in E} c_{hk}x_{hk} \tag{3.11}$$

$$\text{s.t.} \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V,\ (s,t) \in Q \tag{3.12}$$

$$\sum_{s\in W} (b^+_s \lambda^{hk}_{s+} + b^-_s \lambda^{hk}_{s-}) \le x_{hk} \qquad \forall \{h,k\} \in E \tag{3.13}$$

$$\lambda^{hk}_{s+} + \lambda^{hk}_{t-} \ge (f^{st}_{hk} + f^{st}_{kh}) \qquad \forall \{h,k\} \in E, (s,t) \in Q \tag{3.14}$$

$$f^{st}_{hk} \in \{0,1\} \qquad \forall (h,k) \in A,\ (s,t) \in Q \tag{3.15}$$

$$\lambda^{hk}_{s+},\ \lambda^{hk}_{s-} \ge 0 \qquad \forall \{h,k\} \in E,\ s \in W \tag{3.16}$$

$$x_{hk} \ge 0 \qquad \forall \{h,k\} \in E. \tag{3.17}$$

Proof. Start from the semi-infinite flow formulation $HOSE_{sem}$. Consider a single edge $\{h,k\}$ and treat the f variables as parameters. The worst-case value for the capacity x_{hk} to be reserved on edge $\{h,k\}$ is obtained by solving the following

optimization problem:

$$x_{hk} \geq \max \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \tag{3.18}$$

$$(\lambda^{hk}_{s+}) \qquad \sum_{t\in W\setminus\{s\}} d_{st} \leq b^{+}_{s} \qquad \forall s \in W$$

$$(\lambda^{hk}_{s-}) \qquad \sum_{t\in W\setminus\{s\}} d_{ts} \leq b^{-}_{s} \qquad \forall s \in W$$

$$d_{st} \geq 0 \qquad \forall (s,t) \in Q, \tag{3.19}$$

where the λ's are the corresponding dual variables. Since this linear program is feasible and bounded, we can apply a duality transformation similarly to Soyster [61] to obtain the equivalent formulation:

$$x_{hk} \geq \min \sum_{s\in W} (b^{+}_{s}\lambda^{hk}_{s+} + b^{-}_{s}\lambda^{hk}_{s-}) \tag{3.20}$$

$$\lambda^{hk}_{s+} + \lambda^{hk}_{t-} \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall (s,t) \in Q$$

$$\lambda^{hk}_{s+},\ \lambda^{hk}_{s-} \geq 0 \qquad \forall s \in W. \tag{3.21}$$

By replacing constraints (3.18)-(3.19) with (3.20)-(3.21), we obtain a lower bound on the capacity that is required on edge $\{h,k\}$. This substitution immediately leads from $HOSE_{sem}$ to the linear MIP formulation (3.11)-(3.17) after observing that the minimum in (3.20) can be omitted due to the nature of the objective function, which is a non-negatively weighted sum of the variables x_{hk}, and the continuous nature of the x_{hk} variables. In other words, since we have non-negative c_{hk} coefficients in the objective function of $HOSE_{sem}$, minimizing the objective function requires each x_{hk} variable to be pushed down to the lowest possible value it can take while supporting any fesible demand realization. Moreover, given that x_{hk} are continuous variables, (3.20) would be satisfied as equality at optimality. As a result, we can simply remove the minimum in (3.20). □

Remark 1. *Similar to the case mentioned in Soyster [61], $HOSE_{sem}$ is an inexact model where we do not know the pairwise demands at certainty. All we can do is to use a polyhedral set to define possible traffic patterns and we want a design that safeguards the network for any feasible realization. Therefore, we determine the link capacities such that they support the worst cases, i.e. the maximum possible traffic flow on each link. To this end, we follow the treatment in*

Soyster [61] and for each edge $\{h,k\}$, we equally replace the constraint (3.10) with $\sup_{d\in U_{AsymG}} \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \leq x_{hk}$. *Then, the duality transformation mentioned in the proof of Proposition3.3.1 follows since U_{AsymG} is a non-empty compact set.*

Analogously, for the *Sym-G* case we obtain a compact formulation with the same objective function (3.11), the flow conservation constraint (3.12), constraints (3.15), and

$$\begin{aligned}
\sum_{s\in W} b_s \lambda^{hk}_s &\leq x_{hk} && \forall\{h,k\}\in E \\
\lambda^{hk}_s + \lambda^{hk}_t &\geq f^{st}_{hk} + f^{st}_{kh} && \forall\{h,k\}\in E,\ \forall (s,t)\in Q \\
\lambda^{hk}_s &\geq 0 && \forall\{h,k\}\in E,\ \forall s\in W \\
x_{hk} &\geq 0 && \forall\{h,k\}\in E.
\end{aligned} \tag{3.22}$$

As we shall see in Section 3.5, tackling this compact linear MIP formulation with commercial solvers (e.g., Cplex 8.1) yields optimal solutions in reasonable time even for large-size instances.

3.3.1.3 Cutting planes and lower bounds

Some valid inequalities can be easily derived for the formulations given in the preceding paragraphs. These inequalities are useful in cutting plane procedures for numerical solution of large instances as we shall see in Section 3.5.

Let us consider the flow formulation (3.11)-(3.17) of *Asym-G*. For any subset of edges $F \subseteq E$, we use the notation $x(F) = \sum_{\{h,k\}\in F} x_{hk}$ and, similarly, for any subset of arcs $F' \subseteq A$ the notation $f^{st}(F') = \sum_{(h,k)\in F'} f^{st}_{hk}$. For any subset of nodes $S \subset V$, the *cut* $\delta(S)$ is the set of edges with only one end-node in S, i.e., $\delta(S) = \{\{h,k\}\in E : h\in S, k\notin S\}$. For any subset $S\subset V$, the *directed cut* $\delta'(S)$ is the set of arcs with the tail in S and the head in $V\setminus S$.

Let $S\subset V$ be such that there exists a non-empty set of terminal pairs $Q' = \{(s,t)\in Q : s\in S, t\in V\setminus S\}$. Then, for each one of these terminal pairs (s,t),

at least one flow variable associated with an arc of the directed cut $\delta'(S)$ must be non-zero, that is,

$$f^{st}(\delta'(S)) \geq 1. \tag{3.23}$$

In sequel, we obtain the following family of inequalities:

$$\sum_{\{h,k\}\in\delta(S)} (\lambda^{hk}_{s+} + \lambda^{hk}_{t-}) \geq 1 \qquad \forall (s,t) \in Q : |\{s,t\} \cap S| = 1 \tag{3.24}$$

from (3.14) that only involve λ variables and that are satisfied by any feasible solution of *Asym-G*.

Moreover, the capacity $x(\delta(S))$ across the cut $\delta(S)$ must support the demands between all terminal pairs whose endpoints are on different shores. A lower bound is as follows:

$$x(\delta(S)) \geq \sum_{(s,t)\in Q:|\{s,t\}\cap S|=1} d_{st}. \tag{3.25}$$

Given the traffic uncertainty, the right-hand side of (3.25) is not known a priori and $x(\delta(S))$ has to be large enough in the worst case scenario. Since the maximum traffic across a cut $\delta(S)$, denoted by $d(\delta(S))$, amounts to

$$\max \left\{ \sum_{(s,t)\in Q:|\{s,t\}\cap S|=1} d_{st} : d \in U_{\text{AsymG}} \right\},$$

any feasible solution of *Asym-G* must satisfy $x(\delta(S)) \geq d(\delta(S))$. As x_{hk} is defined in constraints (3.13), we can write equivalently:

$$\sum_{\{h,k\}\in\delta(S)} \sum_{s\in W} (b^+_s \lambda^{hk}_{s+} + b^-_s \lambda^{hk}_{s-}) \geq d(\delta(S)). \tag{3.26}$$

The *demand cut-set inequalities* (3.24) and the *capacity cut-set inequalities* (3.26) express two necessary conditions for supporting all valid traffic matrices.

Inequalities (3.24) and (3.26) are not violated by a solution of the linear relaxation of the formulation (3.11)-(3.17), as they are implied by the flow conservation constraints (3.12). However, a *cut formulation* including only the λ variables and

a subset of these inequalities provides a lower bound on *Asym-G*. As for (3.26), we may get rid of the x variables and express the objective function in terms of the λ variables only:

$$\sum_{\{h,k\}\in E} c_{hk} \sum_{s\in W} (b_s^+ \lambda_{s+}^{hk} + b_s^- \lambda_{s-}^{hk}). \tag{3.27}$$

The goal is then to minimize (3.27) subject to inequalities (3.24) and (3.26). A cutting plane procedure for obtaining a lower bound on the optimal value of *Asym-G* takes a small initial set of such inequalities and repeatedly add violated cuts identified through efficient separation procedures. While separating inequalities (3.24) requires solving a maximum-flow problem, and hence is easy, separating inequalities (3.26) can be shown to be NP-hard[2].

In the present chapter we derive dual (lower) bounds on the optimal value by using inequalities (3.24). Consider $G = (V, E)$ and the set of values (b_s^+, b_s^-) $\forall s \in W$. Let Δ denote a subset of all possible triplets $\{s, t, \delta(S)\}$, where $S \subset V$ and $|\{s, t\} \cap S| = 1$. A lower bound is given by the following linear program:

$$\begin{aligned}
(P_{\text{dual}}) \quad \min \quad & \sum_{\{h,k\}\in E} c_{hk} \sum_{s\in W} (b_s^+ \lambda_{s+}^{hk} + b_s^- \lambda_{s-}^{hk}) & \\
\text{s.t.} \quad & \sum_{\{h,k\}\in\delta(S)} (\lambda_{s+}^{hk} + \lambda_{t-}^{hk}) \geq 1 & \forall (s,t,S) \in \Delta \\
& \lambda_{s+}^{hk},\ \lambda_{s-}^{hk} \geq 0 & \forall s \in W,\ \{h,k\} \in E.
\end{aligned}$$

Since this problem does not involve the f variables, it is quickly solved for a reasonable number of inequalities.

Inequalities (3.24) and (3.26) are easily adapted to *Sym-G*. From (3.22) and (3.23) we obtain the following two families of valid inequalities:

$$\sum_{\{h,k\}\in\delta(S)} (\lambda_s^{hk} + \lambda_t^{hk}) \geq 1 \quad \forall\ S \subset V,\ (s,t) \in Q : |\{s,t\} \cap S| = 1$$

[2]Jain et al. [41] provide a polynomial time separation algorithm for a similar problem appeared in a special type of survivable network design problems. However, in order to use their algorithm for (3.26), $d(\delta(S))$ must be a weakly supermodular function of S, which is not the case for our problem.

$$\sum_{\{h,k\}\in\delta(S)} \sum_{s\in W} b_s \ \lambda_s^{hk} \geq d(\delta(S)) \quad \forall \ S \subset V.$$

3.3.2 Robust VPN provisioning problem

Since detailed traffic statistics (reliable average and standard deviation estimates of terminal-to-terminal traffic) are available for existing VPNs, we consider a variant of the VPN provisioning problem which exploits this additional information. Unlike the hose model, where bounds are imposed on the aggregated traffic through each terminal, we adopt a less conservative traffic uncertainty model and consider each demand d_{st} individually. However, our assumptions regarding the valid traffic matrices are very mild: we just assume that we know the range of values that each demand d_{st} can take. Specifically, for each pair of terminals $s, t \in W$, the demand d_{st} is assumed to take values in an interval $[d'_{st} - \hat{d}_{st}, d'_{st} + \hat{d}_{st}]$, where $\hat{d}_{st} \in [0, d'_{st}]$ denotes the non-negative deviation from the nominal value d'_{st}. It is worth emphasizing that we do not make any assumptions concerning the distribution of the traffic demands within the corresponding intervals and concerning the way different demands d_{st} may be interrelated.

For non-negative deviations $\hat{d}_{st}$, the robust VPN provisioning problem in which demands are subject to the above interval uncertainty can be formulated as the following binary integer program (VPN_{int}):

$$\min \sum_{\{h,k\}\in E} c_{hk} \sum_{(s,t)\in Q} (d'_{st} + \hat{d}_{st})(f_{hk}^{st} + f_{kh}^{st}) \tag{3.28}$$

$$\text{s.t.} \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, \ (s,t) \in Q$$

$$f_{hk}^{st} \in \{0,1\} \qquad \forall (h,k) \in A, \ (s,t) \in Q.$$

This model corresponds to the worst case where all demands will be realized at their upper bounds. Hence, just like the nominal problem VPN_{nom} corresponding to the nominal traffic matrix $d' = [d'_{st}]_{s,t\in W}$, this robust version can be solved in polynomial time.

However, it may be overly pessimistic to assume that all demands simultaneously vary within the corresponding uncertainty intervals so as to adversely affect the solution. Therefore, we consider a positive integer parameter Γ, $1 \leq \Gamma \leq |Q|$, which allows us to adjust the trade-off between the VPN robustness and its degree of conservatism as in the general approach for robust discrete optimization problems under data uncertainty proposed in [14, 15]. Whereas minimum and maximum values for point-to-point demand can be obtained through repeated measurements over time, it is difficult to predict when a demand is at its maximum. Although in principle we cannot exclude the possibility that all demands take their peak values simultaneously, traffic statistics show that in general peaks in the demand values are reached at different times. It is then reasonable to limit the conservativeness of the model by adopting a robust approach à la Bertsimas and Sim [14, 15] and by assuming that at any point in time at most Γ of the pairwise demands simultaneously take their worst-case values. In the VPN provisioning setting, the parameter Γ can be interpreted as the margin on the risk that the operator is willing to take for not satisfying the customer requirements over a given period of time.

In order to avoid over-dimensioned VPNs, we minimize the maximum reservation cost while protecting us against the extreme behavior of at most Γ of the demands. More precisely, we consider the following robust VPN provisioning problem *(Rob-G)*:

$$\min \sum_{\{h,k\} \in E} c_{hk} x_{hk} \tag{3.29}$$

$$\text{s.t.} \quad \sum_{k:\{h,k\} \in E} \left(f_{hk}^{st} - f_{kh}^{st} \right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V,\ (s,t) \in Q$$

$$\max_{\{\tilde{Q}\,:\,\tilde{Q} \subseteq Q, |\tilde{Q}| \leq \Gamma\}} \sum_{(s,t) \in Q} d'_{st} (f_{hk}^{st} + f_{kh}^{st}) + \sum_{(s,t) \in \tilde{Q}} \hat{d}_{st} \left(f_{hk}^{st} + f_{kh}^{st} \right) \leq x_{hk} \quad \forall \{h,k\} \in E \tag{3.30}$$

$$f_{hk}^{st} \in \{0,1\} \qquad \forall (h,k) \in A,\ (s,t) \in Q$$

$$x_{hk} \geq 0 \qquad \forall \{h,k\} \in E, \tag{3.31}$$

where $\tilde{Q}$ denotes a subset of Q of cardinality at most Γ. It is worth pointing out that the general probabilistic guarantees derived for the extreme behaviour of more than Γ parameters in [14] are also valid in our case. In other words, even if more than Γ parameters change, the probability of the robust design to be feasible is very high. We also note that choosing Γ equal to the cardinality of Q is equivalent to setting all demand values at their upper bounds, which yields the problem VPN_{int}.

As for *Sym-G* and *Asym-G*, this robust variant of the VPN provisioning problem admits a compact linear MIP formulation. The following result can be viewed as a special case of the first theorem in [14] for general discrete optimization problems.

Proposition 3.3.2. *The robust VPN provisioning problem* (Rob-G) *has the following equivalent linear mixed-integer programming formulation:*

$$\min \sum_{\{h,k\}\in E} c_{hk} x_{hk} \tag{3.32}$$

$$\text{s.t.} \quad \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall i \in V,\ \forall (s,t) \in Q$$

$$\sum_{(s,t)\in Q} d'_{st}(f^{st}_{hk} + f^{st}_{kh}) + \Gamma \pi^0_{hk} + \sum_{(s,t)\in Q} \pi^{st}_{hk} \le x_{hk} \quad \forall \{h,k\} \in E \tag{3.33}$$

$$\pi^0_{hk} + \pi^{st}_{hk} \ge \hat{d}_{st}(f^{st}_{hk} + f^{st}_{kh}) \qquad \forall \{h,k\} \in E,\ \forall (s,t) \in Q \tag{3.34}$$

$$f^{st}_{hk} \in \{0,1\} \qquad \forall (h,k) \in A,\ \forall (s,t) \in Q$$

$$\pi^{st}_{hk} \ge 0 \qquad \forall \{h,k\} \in E,\ \forall (s,t) \in Q$$

$$\pi^0_{hk},\ x_{hk} \ge 0 \qquad \forall \{h,k\} \in E. \tag{3.35}$$

Proof. As in [14, 15], we first consider the constraints whose parameters are subject to uncertainty, namely constraints (3.30). Given a vector $\bar{\mathbf{f}} \in \{0,1\}^{|A|\times|Q|}$, for each $\{h,k\} \in E$, the problem

$$\max_{\{\tilde{Q}\,:\,\tilde{Q}\subseteq Q, |\tilde{Q}|\le\Gamma\}} \sum_{(s,t)\in\tilde{Q}} \hat{d}_{st}\ (\bar{f}^{st}_{hk} + \bar{f}^{st}_{kh}) \tag{3.36}$$

is equivalent to

$$\max \sum_{(s,t)\in Q} \alpha_{st}\ \hat{d}_{st}\ (\bar{f}^{st}_{hk} + \bar{f}^{st}_{kh}) \tag{3.37}$$

$$\text{s.t.} \quad \sum_{(s,t)\in Q} \alpha_{st} \leq \Gamma \tag{3.38}$$

$$\alpha_{st} \in \{0,1\} \qquad \forall (s,t) \in Q. \tag{3.39}$$

The key observation here is that the above problem, viewed as a linear program by relaxing the integrality of α_{st} variables, always has a binary optimal solution as it is a simple knapsack. Therefore, it can be replaced with

$$\max \sum_{(s,t)\in Q} \alpha_{st}\ \hat{d}_{st}\ (\bar{f}^{st}_{hk} + \bar{f}^{st}_{kh}) \tag{3.40}$$

$$\text{s.t.} \quad \sum_{(s,t)\in Q} \alpha_{st} \leq \Gamma \tag{3.41}$$

$$0 \leq \alpha_{st} \leq 1 \qquad \forall (s,t) \in Q, \tag{3.42}$$

which admits the following dual:

$$\min\ \Gamma\pi^0_{hk} + \sum_{(s,t)\in Q} \pi^{st}_{hk} \tag{3.43}$$

$$\text{s.t.} \quad \pi^0_{hk} + \pi^{st}_{hk} \geq \hat{d}_{st}(\bar{f}^{st}_{hk} + \bar{f}^{st}_{kh}) \quad \forall (s,t) \in Q \tag{3.44}$$

$$\pi^0_{hk} \geq 0 \tag{3.45}$$

$$\pi^{st}_{hk} \geq 0 \qquad \forall (s,t) \in Q. \tag{3.46}$$

By strong duality this dual is feasible and bounded and has the same optimal objective function value as the primal problem (3.37)-(3.39).

Substituting (3.43)-(3.46) in *Rob-G* for each link $\{h,k\} \in E$ yields (3.32)-(3.35). □

Unlike the robust discrete optimization problems with $0-1$ variables considered in [14], there is unfortunately strong evidence that the robust VPN provisioning problem is substantially harder to solve than the associated nominal problem, that is VPN_{int} with $\hat{d}_{st} = 0$ for all $(s,t) \in Q$, which can be solved in polynomial time.

Proposition 3.3.3. *The robust VPN provisioning problem* Rob-G *is* NP*-hard.*

Proof. We proceed by polynomial time reduction from the following version of the Satisfiability problem.

> 3-SAT: Given a set of m disjunctive clauses $C_1, \ldots, C_m$ in n Boolean variables $x_1, \ldots, x_n$ and their complements $\overline{x}_1, \ldots, \overline{x}_n$ with exactly 3 literals per clause, does there exist a truth assignment for the variables that satisfies all the clauses?

In other words, the problem is to check the existence of a truth value (TRUE or FALSE, 0 or 1 etc.) assignment for each variable x_j for $j = 1, ..., n$ such that the entire statement composed of the clauses C_i for $i = 1, .., m$ joined using AND, OR, NOT, is TRUE. This problem is known to be NP-complete even when restricted to instances in which each Boolean variable x_j occurs (as x_j or $\overline{x}_j$) in at most 5 clauses ([26]). The construction is similar to the one used to establish that it is NP-complete to decide whether in a given graph k distinct pairs of nodes can be connected with k edge-disjoint paths.

For any such instance of 3-SAT we will construct a special instance of the robust VPN provisioning problem, defined by a graph $G = (V, E)$ with $c_{hk} = 1$ for all $\{h, k\} \in E$, a set of terminals $W \subseteq V$, appropriate nominal demands and deviations between each pair of terminals and $\Gamma = 1$. We will then verify that the former instance of 3-SAT is satisfiable if and only if the latter instance admits a robust VPN of total cost at most $11n + 2m$. In other words, if we can design a VPN with a total cost $11n + 2m$, which is robust to a change in one pairwise demand, then we can determine a truth setting such that each literal appears as x_j or $\overline{x}_j$ and the overall statement is true.

Consider an arbitrary instance of 3-SAT in which each Boolean variable occurs in at most 5 clauses. For each variable x_j in this instance, we consider two terminals s_j and t_j in W connected by two separate parallel paths corresponding, respectively, to the variable x_j and its complement $\overline{x}_j$. Each one of these (s_j, t_j)-paths consists of 11 edges and has 10 intermediate nodes. For each clause C_i, we

consider two other terminals v_i and w_i in W connected by three separate parallel paths that correspond to the three literals in the clause. Each one of these (v_i, w_i)-paths consists of 3 edges and has 2 intermediate nodes. The paths associated with the pairs of terminals (s_j, t_j), $1 \leq j \leq n$, and (v_i, w_i), $1 \leq i \leq m$, must intersect in the appropriate way. For example, if the 3-SAT instance contains a first clause C_{i_1} with the literals x_{j_1}, $\overline{x}_{j_2}$ and x_{j_3}, and a second clause C_{i_2} with the literals $\overline{x}_{j_1}$, x_{j_2} and x_{j_3}, the three separate parallel paths associated with the three literals of each clause are constructed as shown in Figure 3.2. In the figure, paths corresponding to each x_j variable is shown in solid lines whereas the paths in dashed lines correspond to the $\overline{x}_j$ variables. Moreover, paths for each (v_i, w_i) pair are drawn as dotted lines. The first (v_{i_1}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting terminals s_{j_1} and t_{j_1} that corresponds to variable x_{j_1}; the second (v_{i_1}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting s_{j_2} and t_{j_2} that corresponds to $\overline{x}_{j_2}$; the third (v_{i_1}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting s_{j_3} and t_{j_3} that corresponds to x_{j_3}. The three (v_{i_2}, w_{i_2})-paths corresponding to the three literals of the second clause are constructed accordingly. Note that, since the variable x_{j_3} occurs in both clauses, the (s_{j_3}, t_{j_3})-path associated with x_{j_3} (the first one) shares its second edge with the third (v_{i_1}, w_{i_1})-path and its fourth edge with the third (v_{i_2}, w_{i_2})-path. On the other hand, since $\overline{x}_{j_3}$ does not appear in any of the clauses, it does not share any edge with any of the (v_i, w_i) paths.

Since each Boolean variable occurs in at most 5 clauses and each (s_j, t_j)-path consists of 11 edges, we can make sure that each edge of any (s_j, t_j)-path (associated with x_j or $\overline{x}_j$) belongs to at most one (v_i, w_i)-path corresponding to a literal of a clause on the specific graph on Figure 3.2. Note that all paths in G which connect any (s_j, t_j) (respectively (v_i, w_i)) terminal pair but are not (s_j, t_j)-paths (respectively (v_i, w_i)-paths) contain more than 11 (respectively 3) edges.

The special instance of the robust VPN provisioning problem corresponding to the given 3-SAT instance is defined, together with the above graph G and set of terminals $W \subseteq V$, by the nominal demands $d'_{s_j t_j} = 0$ for all j and $d'_{v_i w_i} = 0$ for all i, and the deviations $\hat{d}_{s_j t_j} = 1$ for all j and $\hat{d}_{v_i w_i} = 1$ for all i. The nominal

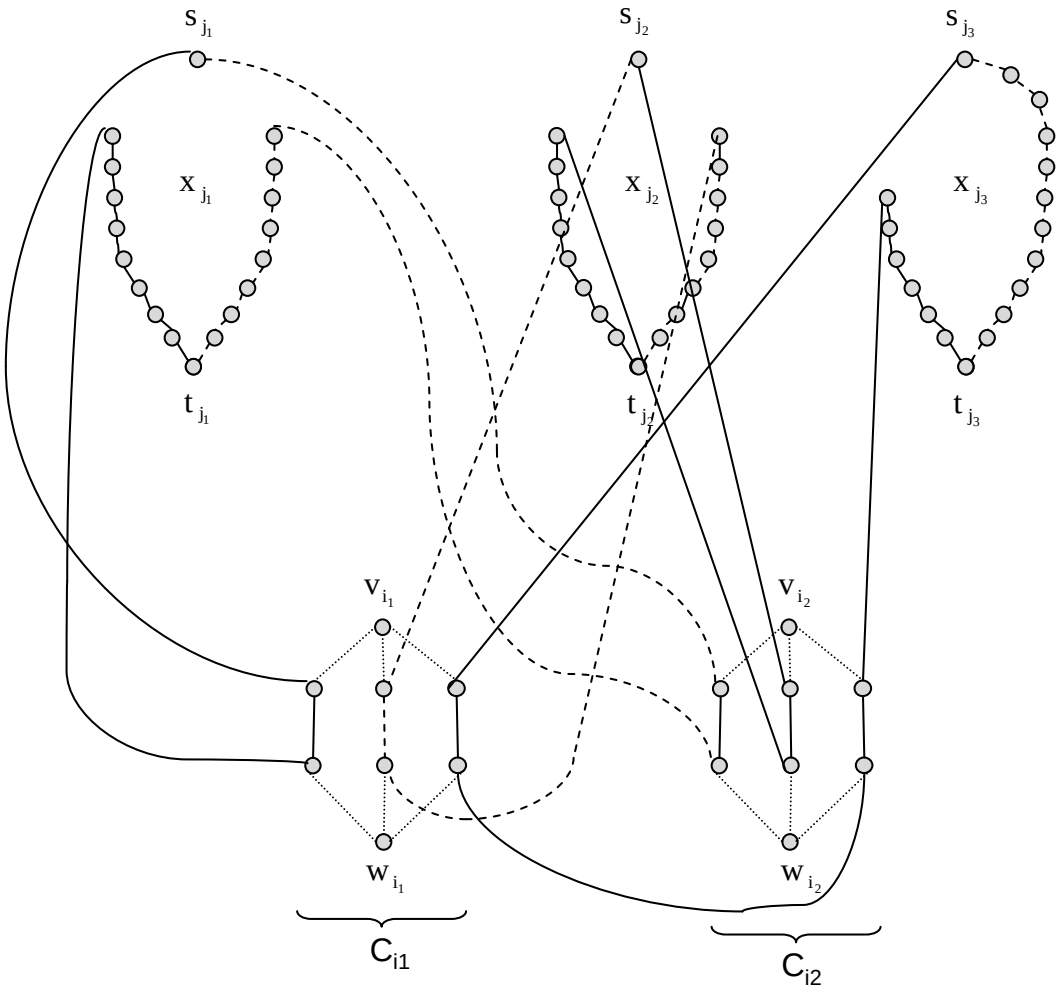

Figure 3.2: Graph of the special instance of the VPN provisioning problem associated with a 3-SAT instance with the clauses $C_{i_1} = (x_{j_1} \vee \bar{x}_{j_2} \vee x_{j_3})$ and $C_{i_2} = (\bar{x}_{j_1} \vee x_{j_2} \vee x_{j_3})$.

demands and deviations for all other pairs of terminals in W are set to zero.

Since $\Gamma = 1$ and by construction each (v_i, w_i)-path shares exactly one edge with one of the (s_j, t_j)-paths, asking whether there exists a robust VPN with total reservation cost at most $11n + 2m$ amounts to deciding whether it is possible to route a unit of flow from s_j to t_j (i.e., to select one of the two alternative 11-edge (s_j, t_j)-paths) for every j, $1 \leq j \leq n$, and a unit of flow from v_i to w_i for every i, $1 \leq i \leq m$ (i.e., to select one of the three alternative 3-edge (v_i, w_i)-paths) so that for each i the selected (v_i, w_i)-path shares exactly one edge with a selected (s_j, t_j). Note that, due to unit costs, the total reservation cost cannot be smaller than $11n + 3m - m = 11n + 2m$. Indeed, a path with at least 11 (respectively 3) edges is needed to connect each one of the n (respectively m) terminal pairs (s_j, t_j) (respectively (v_i, w_i)) and each selected (v_i, w_i)-path shares at most one edge with one of the selected (s_j, t_j)-paths. In other words, for $\Gamma = 1$ and the given d' and $\hat{d}$ vectors, the network should be ready to carry at most 1 unit of demand between at most one of (s_j, t_j) or (v_i, w_i) pair for $1 \leq j \leq n$ and $1 \leq i \leq m$. Hence, we must reserve a unit capacity on all links of each of the $n + m$ paths.

Given any truth assignment for the 3-SAT instance, it is straightforward to select appropriate routing paths for the n pairs of terminals (s_j, t_j) and m pairs of terminals (v_i, w_i) based on the truth values of the variables and literals (at least one literal in each clause is true). This is because the value of each x_j variable shows which of the two paths between each (s_j, t_j) pair will be designed. Moreover, since each (v_i, w_i) pair shares exactly one edge with a (s_j, t_j)-path, the values of the variables also designate the m paths in use for all (v_i, w_i) pairs. Conversely, it is easy to derive from any such $n + m$ paths (where each selected (v_i, w_i)-path shares exactly one edge with one of the selected (s_j, t_j)-paths) a truth assignment for the original instance of 3-SAT.

Since demands are assumed to be routed along single paths, the reduction still holds for non-zero nominal demands d' for all (s_j, t_j) and (v_i, w_i) terminal pairs. For instance, if $d'_{s_j t_j} = 1$ and $d'_{v_i w_i} = 1$, the question is whether there exists a robust VPN of total cost at most $22n + 4m$. □

Note that an important difference with respect to the $0-1$ robust discrete optimization problems considered in [14] lies in the fact that here each uncertain parameter $\hat{d}_{st}$ multiplies several binary variables f^{st}_{hk}, see (3.28) and (3.30), instead of a single one.

In Section 3.5, we report computational results obtained by tackling this compact linear MIP formulation for medium-to-large-size instances with the Cplex 8.1 MIP solver.

3.3.2.1 Cutting planes

Valid inequalities similar to those in Section 3.3.1.3 can also be derived for *Rob-G*. Consider the relaxation of (3.36) equivalent to (3.40)-(3.42):

$$\begin{aligned}
& \max \sum_{(s,t)\in Q} d_{st}(\bar{f}^{st}_{hk} + \bar{f}^{st}_{kh}) \\
\text{s.t.} & \\
(\pi^0_{hk}) \qquad & \sum_{(s,t)\in Q} \frac{d_{st} - d'_{st}}{\hat{d}_{st}} \leq \Gamma \\
(\sigma^{st}_{hk}) \qquad & -d_{st} \leq -d'_{st} \qquad \forall (s,t) \in Q \\
(\rho^{st}_{hk}) \qquad & d_{st} \leq d'_{st} + \hat{d}_{st} \qquad \forall (s,t) \in Q,
\end{aligned}$$

where π^0_{hk}, σ^{st}_{hk}, and ρ^{st}_{hk} are the corresponding dual variables. It is easy to show that constraints (3.33) and (3.34) can be replaced by

$$\left(\Gamma + \sum_{(s,t)\in Q} d'_{st}/\hat{d}_{st}\right)\pi^0_{hk} + \sum_{(s,t)\in Q} \left((d'_{st} + \hat{d}_{st})\rho^{st}_{hk} - d'_{st}\sigma^{st}_{hk}\right) \leq x_{hk} \qquad \forall \{h,k\} \in E$$

$$\pi^0_{hk}/\hat{d}_{st} - \sigma^{st}_{hk} + \rho^{st}_{hk} \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

$$\pi^0_{hk},\ \sigma^{st}_{hk},\ \rho^{st}_{hk} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

and that the two types of valid inequalities for *Rob-G* are:

$$\sum_{\{h,k\}\in\delta(S)} (\pi^0_{hk}/\hat{d}_{st} - \sigma^{st}_{hk} + \rho^{st}_{hk}) \geq 1 \qquad \forall\ S \subset V,\ (s,t) \in Q : |\{s,t\} \cap S| = 1$$

$$\sum_{\{h,k\}\in\delta(S)} \left(\left(\Gamma + \sum_{(s,t)\in Q} d'_{st}/\hat{d}_{st} \right) \pi^0_{hk} + \sum_{(s,t)\in Q} \left((d'_{st} + \hat{d}_{st})\rho^{st}_{hk} - d'_{st}\sigma^{st}_{hk} \right) \right) \geq d(\delta(S)) \quad \forall\ S \subset V.$$

3.3.3 Path formulations

The flow-based models of the preceding sections quickly become too large for medium-to-large network design instances. The resulting test problems are usually too time-consuming for numerical solution by off-the-shelf MIP solvers. An alternative approach is to adopt path-based formulations suitable for column generation algorithms.

We now consider path variables instead of flow variables. Moreover, we let P denote the set of all possible oriented paths between any two given nodes in W, P_{hk} be the set of all paths in P containing the edge $\{h,k\} \in E$, and P^{st} be the set of paths between two demand nodes s and t. Then, for each path $p \in P$, we consider a binary variable z_p that is equal to 1 if and only if the path p, implicitly defined between two nodes s and t, is used to satisfy the demand d_{st}. Notice that these variables are binary as we require unsplittable routing.

Consequently, the initial semi-infinite path formulation for *Asym-G* can be expressed as follows:

$$\begin{aligned}
&\min \sum_{\{h,k\}\in E} c_{hk}x_{hk} && && \\
\text{s.t.} \quad &\sum_{p\in P^{st}} z_p \geq 1 && \forall (s,t) \in Q && (3.47)\\
&\sum_{(s,t)\in Q} d_{st} \sum_{p\in P^{st}\cap P_{hk}} z_p \leq x_{hk} && \forall d \in U_{\text{AsymG}},\ \{h,k\} \in E && (3.48)\\
&z_p \in \{0,1\} && \forall p \in P && \\
&x_{hk} \geq 0 && \forall \{h,k\} \in E. &&
\end{aligned}$$

Constraints (3.47) ensure demand satisfaction by imposing that at least one path is used between each terminal pair, while constraints (3.48) define sufficient capacity on each edge $\{h,k\}$ to support all traffic matrices in the uncertainty set U_{AsymG}. Using the same arguments in the proof of Proposition 3.3.1 in Section 3.3.1.2, we obtain the linear mixed-integer path-based formulation for *Asym-G* as follows:

$$\min \sum_{\{h,k\}\in E} c_{hk}x_{hk} \tag{3.49}$$

$$\text{s.t.} \quad \sum_{p\in P^{st}} z_p \geq 1 \qquad \forall\,(s,t)\in Q \tag{3.50}$$

$$\sum_{s\in W}(b_s^+\lambda_{s+}^{hk} + b_s^-\lambda_{s-}^{hk}) \leq x_{hk} \quad \forall\{h,k\}\in E \tag{3.51}$$

$$\lambda_{s+}^{hk} + \lambda_{t-}^{hk} \geq \sum_{p\in P^{st}\cap P_{hk}} z_p \quad \forall\,(s,t)\in Q,\ \{h,k\}\in E \tag{3.52}$$

$$z_p \in \{0,1\} \qquad \forall\,(s,t)\in Q,\ p\in P^{st} \tag{3.53}$$

$$\lambda_{s+}^{hk},\ \lambda_{s-}^{hk} \geq 0 \qquad \forall\,\{h,k\}\in E,\ s\in W \tag{3.54}$$

$$x_{hk} \geq 0 \qquad \forall\,\{h,k\}\in E. \tag{3.55}$$

Similarly, for *Sym-G* we have the following path-based robust formulation:

$$\min \sum_{\{h,k\}\in E} c_{hk}x_{hk}$$

$$\text{s.t.} \quad \sum_{p\in P^{st}} z_p \geq 1 \qquad \forall\,(s,t)\in Q$$

$$\sum_{s\in W} b_s\ \lambda_s^{hk} \leq x_{hk} \qquad \forall\{h,k\}\in E$$

$$\lambda_s^{hk} + \lambda_t^{hk} \geq \sum_{p\in P^{st}\cap P_{hk}} z_p \quad \forall\,(s,t)\in Q,\ \{h,k\}\in E$$

$$z_p \in \{0,1\} \qquad \forall\,(s,t)\in Q,\ p\in P^{st}$$

$$\lambda_s^{hk} \geq 0 \qquad \forall\,\{h,k\}\in E,\ s\in W$$

$$x_{hk} \geq 0 \qquad \forall\,\{h,k\}\in E,\ s\in W.$$

Finally, a linear path-based formulation is obtained for *Rob-G* as follows:

$$
\begin{aligned}
& \min \sum_{\{h,k\}\in E} c_{hk}x_{hk} & \\
\text{s.t.} \quad & \sum_{p\in P^{st}} z_p \geq 1 & \forall\,(s,t)\in Q \\
& \sum_{(s,t)\in Q} d'_{s,t} \sum_{p\in P^{st}\cap P_{hk}} z_p + \Gamma\pi^0_{hk} + \sum_{(s,t)\in Q} \pi^{st}_{hk} \leq x_{hk} & \forall\{h,k\}\in E \\
& \pi^{st}_{hk} \geq \hat{d}_{st} \sum_{p\in P^{st}\cap P_{hk}} z_p - \pi^0_{hk} & \forall\,\{h,k\}\in E,\ (s,t)\in Q \\
& z_p \in \{0,1\} & \forall\,(s,t)\in Q,\ p\in P^{st} \\
& \pi^{st}_{hk} \geq 0 & \forall\{h,k\}\in E,\ (s,t)\in Q \\
& \pi^0_{hk},\ x_{hk} \geq 0 & \forall\{h,k\}\in E,\ (s,t)\in Q.
\end{aligned}
$$

The above path-based formulations are equivalent to their flow-based counterparts in Proposition 3.3.1 and Proposition 3.3.2, respectively.

3.4 An exact solution method

Since practical VPNs can contain hundreds of terminals in networks with thousands of nodes and our flow formulations of *Asym-G*, *Sym-G*, and *Rob-G* are not viable for networks of that size, we use the more flexible path formulations, which we have described in Section 3.3.3, and adopt a column generation approach. As we are facing linear mixed-integer programs, the path formulations are tackled within a branch-and-price framework discussed in Section 3.4.1. To evaluate the quality of the upper bounds found by the branch-and-price algorithm, we devise a cutting plane procedure which provides lower bounds. This procedure, based on the discussion in Sections 3.3.1.3 and 3.3.2.1, and described below, also helps to deal with the well-known *tailing off* effect (i.e., the generation of spurious columns –paths– that do not improve the objective function value of the linear relaxation), which can substantially affect the performance of column generation. Cutting planes are not used throughout the branch-and-price algorithm, but only in the initial column generation phase performed at the root node: after

each pricing iteration, we run one iteration of the cutting plane procedure.

3.4.1 A branch-and-price and cutting plane algorithm

To take advantage of the path formulations and solve large practical instances of *Asym-G*, *Sym-G*, and *Rob-G*, we combine column generation and branch-and-bound. This joint approach, known as *branch-and-price*, has been introduced by Barnhart et al. [10] to solve large integer programs; an application to a multi-commodity flow problem with integer flow variables can be found in [9]. Although in the following we only describe this approach for *Asym-G*, the versions for *Sym-G* and *Rob-G* are easily derived.

We start from a path formulation (3.49)-(3.55) with a small set P_0^{st} of paths for each terminal pair (s,t). At least one path per terminal pair is needed to ensure feasibility, but we take the K shortest paths from s to t with respect to the edge costs c_{hk}. Relaxing integrality yields the *Restricted Linear Problem* (RLP) that we solve through the primal simplex method, retrieving the values of the dual variables $\bar{\sigma}_{st}$, $\bar{\xi}_{hk}$, $\bar{\eta}_{hk}^{st}$ of constraints (3.50), (3.51), and (3.52), respectively.

Consider a terminal pair $(s,t) \in Q$. A variable z_p corresponding to a path $p \in P^{st}$ has reduced cost $-\bar{\sigma}_{st} + \sum_{\{h,k\} \in p} \bar{\eta}_{hk}^{st}$. All variables included in RLP have zero or positive reduced cost after the primal simplex algorithm is called, but there may exist a z_p, not included in the current formulation, with a negative reduced cost. Seeking one such variable for each terminal pair $(s,t) \in Q$ means solving the so-called *pricing problem*. We do not enumerate all paths in the set of remaining paths $P^{st} \setminus P_0^{st}$, as its cardinality may be exponential in the number of vertices. Instead, we solve a shortest path problem on a directed graph $G_\eta(s,t)$ whose arcs (h,k) correspond to edges in G and have cost $\bar{\eta}_{hk}^{st}$. If the length of such a shortest path p is smaller than $\bar{\sigma}_{st}$, then z_p has negative reduced cost and can be included in RLP. The column generation procedure is summarized in Procedure 3.1.

The column generation procedure is applied until no new path with negative

Procedure 3.1 Column generation procedure

Input: undirected graph $G = (V, E)$, traffic polytope U_{AsymG}, cost vector $\vec{c}$;
Output: optimal solution for RLP;

Initialize:
  Find an initial set P_0^{st} of K shortest paths for every $(s,t) \in Q$;
  Obtain RLP by relaxing the integrality constraints;
Optimize: Solve RLP, obtaining a primal solution $(\bar{x}_{hk}, \bar{\lambda}_{s+}^{hk}, \bar{\lambda}_{s-}^{hk}, \bar{z}_p)$ and the associated dual solution $(\bar{\sigma}_{st}, \bar{\xi}_{hk}, \bar{\eta}_{hk}^{st})$;
Price:
for all $(s,t) \in Q$ **do**
  Build an auxiliary undirected graph $G_{\bar{\eta}}(s,t)$ such that the length of each edge is $\bar{\eta}_{hk}^{st}$;
  Find the shortest path $\tilde{p}$ in $G_{\bar{\eta}}(s,t)$ and let the length of $\tilde{p}$ be $l_{\bar{\eta}}^*(s,t)$;
  If $l_{\bar{\eta}}^*(s,t) < \bar{\sigma}_{st}$, then add the related variable $z_{\tilde{p}}$ to RLP;
if at least one variable is added **then**
  Go to *Optimize*;
else
  Dual feasibility is obtained and RLP is solved.

reduced cost is found. Upon termination the current solution is an optimal solution of RLP. If one or more z_p variables are fractional, we divide the RLP by applying a branching step. The resulting subproblems are relaxed and solved with the technique described above, and so are all subproblems corresponding to nodes in the branch-and-bound tree.

It is worth pointing out that branching on one variable z_p may unbalance the branch-and-bound tree, or even make the algorithm loop: if a variable z_p is fixed to zero but the pricing procedure finds a path $p' \equiv p$ with negative reduced cost, a variable $z_{p'}$ is added even if path p is forbidden. Therefore, the following branching rule is used at a branch-and-bound node N_k with at least one fractional $z_{\check{p}}$ variable related to a terminal pair (s,t). We choose an edge $\check{e}$ contained in $\check{p}$, then create two new nodes N_{k+1} and N_{k+2}, such that all path variables in N_{k+1} for terminal pair (s,t) associated with paths containing $\check{e}$ are set to zero, while in N_{k+2} their sum must be equal to one. Once the constraints

$$\sum_{p \in P^{st}: \check{e} \in p} z_p = 0 \qquad \text{and} \qquad \sum_{p \in P^{st}: \check{e} \in p} z_p = 1$$

are added to the subproblems associated to N_{k+1} and respectively N_{k+2}, the

pricing problem is solved by a shortest path computation which takes into account the branching constraints at upper levels of the branch-and-bound tree.

As a result of a major shortcoming of column generation techniques, the *tailing off* effect, several columns with negative reduced cost are generated even when an optimal solution is reached, requiring unnecessary extra running time. We have used the cutting planes (3.24) described in Section 3.3 to obtain a lower bound and close the gap with respect to the upper bound given by column generation.

We consider a combined branch-and-price and cutting plane algorithm, referred to as BPC, where an iteration of the cutting plane procedure is run after each pricing iteration of the column generation in the root node. For each terminal pair $(s,t) \in Q$, a directed cut $\delta'(S)$ is sought such that $\sum_{(h,k)\in\delta'(S)}(\lambda_{s+}^{hk} + \lambda_{t-}^{hk})$ is minimum, which amounts to solving the maximum (s,t)-flow problem on a graph whose edges $\{h,k\}$ have capacity $(\lambda_{s+}^{hk} + \lambda_{t-}^{hk})$. If the max-flow obtained is less than one, the cutting plane is inserted.

We have observed a dramatic improvement by adding a limited number of cutting planes, whose separation is equivalent to the max-flow problem. Using the notation of Section 3.3.1.3, the cutting plane procedure is outlined in Procedure 3.2.

Procedure 3.2 Cutting plane procedure

$\Delta \leftarrow \emptyset$; $k \leftarrow 0$; *feasible* $\leftarrow$ TRUE;
Solve (P_{dual}), obtaining values $\tilde{\lambda}_{s+}^{hk}$ and $\tilde{\lambda}_{s-}^{hk}$ for all $\{h,k\} \in E$ and $s \in W$;
for all $(s,t) \in Q$ **do**
    Let $\tilde{G}_{st} = (V,A)$ be an auxiliary graph where $A = \{(h,k) : \{h,k\} \in E\}$
    $u_{hk} = u_{kh} = \tilde{\lambda}_{s+}^{hk} + \tilde{\lambda}_{t-}^{hk}$ for all $(h,k) \in A$;
    Solve the maximum-flow/minimum-cut problem over $\tilde{G}_{st} = (V,A)$ with capacities u_{hk} and let $\delta(S)$ be a minimum capacity cut;
    if the capacity of $\delta(S)$ is smaller than 1 **then**
        $\Delta \leftarrow \Delta \cup (s,t,S)$ and *feasible* $\leftarrow$ FALSE;
if *feasible* = FALSE **then**
    Go to *Solve*;
else
    Stop.

3.5 Computational results

We have tested the compact linear MIP formulations and our BPC algorithm for *Sym-G*, *Asym-G*, and *Rob-G* on medium-to-large-size network topologies. The following instances[3] have been considered:

bhv3/6/13: instances of a multi-commodity flow problem studied in [9];

arpanet, eon, latadl, nsf, pacbell, toronto, usld: topologies of well-known backbone networks found in the IEEE literature;

res1/5/6/7/8/9, at-cep, ny-cep, nor-sun: instances of different flow problems of multi-commodity nature found at:
`http://www.di.unipi.it/di/groups/optimize/Data/MMCF.html#Rsrv`;

stein1/2/3/4: a set of Steiner tree problem instances with 50 and 75 nodes, available at the web page:
`http://people.brunel.ac.uk/ mastjjb/jeb/orlib/steininfo.html`;

n45, n49, n147: general multi-commodity flow instances;

g200: a network topology with 200 nodes and 914 links [1];

t3-X, t4-X: a set of backbone networks with 250 and 304 nodes randomly generated with the `gt-itm` software
(`http://www.cc.gatech.edu/projects/gtitm/gt-itm/`).

Since single VPNs are considered, the set W of terminals has been randomly selected as a small subset of V with a density $|W|/|V|$ of approximately 15%. The values of b_s, b_s^+, b_s^-, d'_{st}, and $\hat{d}_{st}$ have been generated based on realistic random traffic matrices.

All tests have been carried out with a Pentium Xeon 2.8 GHz processor, running under Linux with 2 GB of memory available. The BPC algorithm has

[3] All instances are available from: `ftp://ftp.elet.polimi.it/users/Pietro.Belotti/mcf/vpn` in a pseudo-DIMACS format and in AMPL data format.

been implemented in C++ using the open-source COIN/Bcp software framework ([37]) and Cplex 8.1 as LP solver, while Cplex 8.1 MIP solver has been used for our compact formulations. Initially we generate, for each terminal pair (s, t), K shortest paths from s to t through the HREA algorithm described in [42].

The computational results are summarized in five tables that include the following information:

- the instance name and the network parameters (the cardinalities of V, E, and of the set of terminals $W \subseteq V$),
- the time (t_{root}) spent by column generation in the root node, i.e., the time required to solve the relaxed problem, starting with $K = 5$ paths for each terminal pair,
- the time (t_{tot}) spent in total by the BPC algorithm,
- the time (t_{cf}) spent by the Cplex 8.1 MIP solver on the linear flow-based compact formulation,
- #paths and #cuts: the number of paths and cuts generated by the pricing and the cutting plane iterations.

Also note that all times are expressed in seconds.

A time limit of two hours has been given both for the BPC algorithm and for tackling the compact flow-based formulations with the Cplex 8.1 MIP solver. If the time limit is reached before obtaining an optimal solution, we report in brackets the gap between the best feasible solution and the best lower bound that have been found. A "–" indicates that no feasible solution was found (in the t_{tot} column it means that the branch-and-bound part was not performed). The label "*mem*" indicates that the problem could not be solved because of excessive memory requirements, namely more than 2GB of RAM. In the BPC algorithm, if the time limit is reached while solving the linear relaxation, the gap is given in brackets in the t_{root} column.

According to Table 3.1, the small-size *Sym-G* instances (with up to 40 nodes) are solved very rapidly and the performance of Cplex 8.1 on the compact linear MIP formulation and of the BPC algorithm are comparable in most cases.

Table 3.1: Results for small *Sym-G* instances: BPC algorithm and the compact linear MIP formulation solved with Cplex.

name	$\|V\|$	$\|E\|$	$\|W\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf}
arpanet	24	50	10	11.03	11.03	1219	1055	1.97
at-cep	15	22	6	0.02	0.02	211	100	0.03
bhv3	29	62	15	2.65	–	1695	1498	8.34
bhv6	27	39	15	0.14	0.03	1123	648	0.13
bhv13	29	36	13	0.48	0.04	1027	1212	0.86
cost239	11	22	5	0.02	0.02	129	46	0.07
eon	19	37	15	0.49	0.49	1272	714	1.28
latadl	39	86	17	397.27	397.27	3917	4259	219.55
metro	11	42	5	0.01	0.01	126	44	0.26
njlata	11	23	8	0.14	0.14	405	212	0.33
nor-sun	27	51	13	70.53	70.53	2019	1604	64.24
nsf	14	21	10	0.45	0.45	672	422	0.86
ny-cep	16	49	9	0.71	0.71	573	354	0.75
pacbell	15	21	7	0.09	0.09	294	176	0.16
toronto	25	55	11	2.53	2.53	1026	887	6.27
usld	28	45	15	7.55	7.55	1753	1496	4.17

The computational results obtained for larger *Sym-G* instances are reported in Table 3.2.

Our compact flow-based formulation turns out to be very tight so as to yield the optimal solution within less than a minute even for the large problem *n147*. For larger instances, however, it pays to develop a specialized combined branch-and-price and cutting plane method because the compact linear MIP formulation leads to excessive memory requirements. Our BPC algorithm, which performs better on *n147* and on a few smaller instances such as *n45* and *stein2*, allows us to solve *g200* and *t3-3* optimally within the two hour time limit and to tackle larger instances. Unlike other experiments, for *t3-0* we have set $K = 20$ (instead of $K = 5$) and let the BPC method run for 252,286.68 sec. The gap is in fact reduced to 3.24% after 79,293.58 sec. and then progressively to 0.95%. Tuning

Table 3.2: Results for medium-to-large-size *Sym-G* instances: BPC algorithm and the compact linear MIP formulation solved with Cplex.

name	$\|V\|$	$\|E\|$	$\|W\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf}
res1	45	63	15	36.62	43.39	2843	3533	32.41
res5	44	67	18	174.21	174.21	3895	4641	58.78
res6	44	60	14	93.64	93.64	2820	2641	6.63
res7	44	62	18	2920.21	2920.21	11341	10019	37.68
res8	50	79	19	184.27	184.27	4860	6406	92.71
res9	50	76	23	1048.50	74.24	11467	12620	107.16
stein1	50	100	17	40.99	40.99	2778	2657	43.65
stein2	50	63	20	1.46	1.46	2488	2586	1.48
stein3	75	94	32	97.07	97.07	8192	16084	61.88
stein4	75	150	33	3654.09	3654.09	11124	16411	3443.33
n45	45	63	20	5.89	6.51	3187	4512	38.62
n49	49	57	14	1.03	1.03	1357	1378	0.98
n147	147	265	37	18.80	18.80	7668	8211	52.72
g200	200	914	31	58.01	5986	6546	63.42	*mem*
t3-0	250	444	52	(0.95%)	–	97025	246679	*mem*
t3-1	250	456	53	(17.24%)	–	33164	141516	*mem*
t3-2	250	469	42	(1.30%)	–	44395	220123	*mem*
t3-3	250	446	50	6959.42	6959.42	48190	363584	*mem*
t3-4	250	465	39	(0.71%)	–	40939	122825	*mem*
t4-0	304	453	55	(0.02%)	–	42524	293763	*mem*
t4-1	304	457	63	(7.69%)	–	28710	114345	*mem*
t4-2	304	454	58	(19.30%)	–	28335	96122	*mem*
t4-3	304	458	67	(26.88%)	–	33589	133557	*mem*
t4-4	304	468	63	(0.82%)	–	43349	275168	*mem*

the value of K and allowing for more than two hours of computing time, we can also obtain gaps below a few percentage points for the other large instances. Note that, for all instances which have been solved to optimality, an optimal solution is already found at the root node of the branch-and-price tree. The BPC algorithm stops after the initial column generation phase because the solution found has the same value as the optimal integer solution. In very few cases branch-and-bound steps are performed (and the branch-and-price time taken after the RLP is solved, $t_{\text{tot}} - t_{\text{root}}$, is negligible in most cases) and they do not improve the lower bound. Since the number of paths that are generated during the column generation phase is moderate with respect to the number of terminal pairs, the approach is viable even for larger instances. The insertion of cutting planes has a dramatic effect in limiting the tailing off of the objective function value when additional paths are added, thus helping to close the gap within a very short computing time. Notice that the number of cuts separated at the root node, is the same order of magnitude as the number of paths.

Another interesting result emerged from preliminary experiments performed on small, randomly generated *Sym-G* instances. We have observed that for most instances even the linear relaxation has an integer solution, i.e., all flow variables f are either 0 or 1; in the few cases where at least one f variable is fractional, restoring the integrality constraints gives an integer solution with the same cost. Moreover, for most *Sym-G* instances the edges $\{h, k\}$ with positive capacity x_{hk} formed a tree; in the remaining cases, imposing a tree structure on the solution, i.e., solving the related *Sym-T* problem, yielded a solution with the same cost. Thus, our experimental results support the following conjecture:

Sym-G always admits an optimal tree solution.

The same conjecture was also made by Erlebach and Rüegg [23] and it has been proved by Hurkens et al. [38] for networks consisting of a single cycle. Note that the situation for *Asym-G* differs slightly: most instances have an integer optimum with the same cost as the linear relaxation optimal solution, but we have found some instances with an integer optimum with larger cost.

Computational results for *Asym-G* are reported in Table 3.3.

The compact flow-based formulation is remarkably tight and effective also in this case. All mid-size instances can be solved to optimality in a few minutes with the Cplex 8.1 MIP solver, except for *stein4* for which no integer solution could be found within the time limit. Among the large networks, only *t3-4* could be solved to optimality, whereas the remaining *tX-X* and *g200* could not be solved due to excessive memory requirements. In general, asymmetric instances appear to be more challenging than symmetric ones for the BPC approach. Since for 9 out of the 40 instances considered the gap remains large after reaching the time limit, we do not report BPC computing times.

It has still to be determined whether the substantial difference observed for several instances is mainly due to the remarkable quality of the compact linear MIP formulation or to a relevant margin for improving the BPC algorithm, by including for instance efficient memory management procedures that purge unused paths/cuts and λ variables. However, we should point out that passing from the symmetric to the asymmetric hose models renders the problem significantly harder. This is indeed the case for *Sym-T* and *Asym-T*. While *Sym-T* can be solved by repeated application of shortest path algorithms, *Asym-T* is strongly NP-hard. Concerning the behaviour of the branch-and-price algorithm, we should notice that the number of λ variables in the *Asym-G* MIP formulation (3.11)-(3.17) is twice the number of λ variables in the *Sym-G* formulation. We believe that this increase in the number of variables negatively affects the pricing and separation phases of the BPC algorithm. We have observed this in a few numerical examples that we did not report in the tables because the running times were worse than with the compact formulation. Let us consider the three examples *res1*, *stein1*, and *n147*. For the *Sym-G* case from Table 3.2, we see that *res1* was solved to optimality by generating 2843 paths and 3533 cuts. The same statistics when we pass to *Asym-G* uncertainty problems are 4972 and 5952, respectively. For *stein1*, in the *Sym-G* case the number of paths and cuts were 2778 and 2657, respectively. Passing to *Asym-G*, these numbers become 5628 and 17225, respectively. Finally, in the *n147* case, we have 7668 paths and 8211 cuts in the *Sym-G* case and 9823 paths and 35629 cuts for *Asym-G*. This seems to be

Table 3.3: Results for *Asym-G* instances: compact linear MIP formulation solved with Cplex.

name	$\|V\|$	$\|E\|$	$\|W\|$	t_{cf}
arpanet	24	50	10	6.34
at-cep	15	22	6	0.05
bhv3	29	62	15	7.60
bhv6	27	39	15	0.29
bhv13	29	36	13	1.37
cost239	11	22	5	0.04
eon	19	37	15	1.44
latadl	39	86	17	471.06
metro	11	42	5	0.40
njlata	11	23	8	1.02
nor-sun	27	51	13	38.00
nsf	14	21	10	1.62
ny-cep	16	49	9	1.37
pacbell	15	21	7	0.09
toronto	25	55	11	29.14
usld	28	45	15	29.02
res1	45	63	15	37.53
res5	44	67	18	64.14
res6	44	60	14	37.87
res7	44	62	18	39.79
res8	50	79	19	215.50
res9	50	76	23	138.01
stein1	50	63	20	1.91
stein2	50	100	17	96.40
stein3	75	94	32	113.49
stein4	75	150	33	–
n45	45	63	20	102.47
n49	49	57	14	1.13
n147	147	265	37	48.99
t3-4	250	465	39	720.06

a pattern for almost all test problems.

The results obtained for *Rob-G* are summarized in Tables 3.4 and 3.5.

Table 3.4: Results for small-size instances of the *robust* VPN provisioning problem: BPC algorithm and the compact linear MIP formulation solved with Cplex.

name	$\|V\|$	$\|E\|$	$\|W\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf} (gap)
arpanet	24	50	10	0.14	0.21	463	178	(6.88%)
at-cep	15	22	6	0.03	0.04	172	79	0.06
bhv3	29	62	15	0.25	0.35	1059	411	158.50
bhv6	27	39	15	0.33	0.47	1139	414	3.52
bhv13	29	36	13	0.21	0.30	787	306	0.82
cost239	11	22	5	0.02	0.03	108	39	0.04
eon	19	37	15	0.44	0.57	1120	597	3337.24
latadl	39	86	17	1.74	2.14	1532	1028	(9.24%)
metro	11	42	5	0.14	0.15	278	152	0.45
njlata	11	23	8	0.05	0.08	292	156	56.72
nor-sun	27	51	13	0.62	0.74	974	715	686.27
nsf	14	21	10	0.06	0.09	479	172	7.45
ny-cep	16	49	9	0.24	0.29	453	290	7.15
pacbell	15	21	7	0.03	0.05	222	82	1.16
toronto	25	55	11	0.21	0.31	564	219	(6.43%)
usld	28	45	15	0.39	0.54	1076	413	(2.75%)

Here Γ is taken equal to 15% of the total number of terminal pairs, but other intermediate values yield similar results. Although the robust VPN provisioning problem leads to less conservative VPNs by exploiting the available traffic statistics, Tables 3.4 and 3.5 indicate that the corresponding compact linear MIP formulation is much harder to tackle with Cplex 8.1 than those for *Asym-G* and *Sym-G*. However, the BPC algorithm is remarkably effective in solving *Rob-G* instances and it compares very favorably with Cplex 8.1 even for small-size instances. As happens with *Sym-G*, the compact flow-based formulation for *Rob-G* requires more memory than the BPC algorithm in which paths and cuts are dynamically generated starting from small initial sets. For all instances larger than *n147*, the compact formulations do not fit within 2 GB of RAM whereas the path and cut formulations, which have a potentially exponential size, lead to optimal solutions in less than 2 hours. Note that only a limited number of paths and cuts

Table 3.5: Results for medium-to-large-size instances of the *robust* VPN provisioning problem: BPC algorithm and the compact linear MIP formulation solved with Cplex.

name	$\|V\|$	$\|E\|$	$\|W\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf} (gap)
res1	45	63	15	0.91	1.12	1240	810	(1.53%)
res5	44	67	18	1.26	1.59	1615	895	1249.61
res6	44	60	14	0.56	0.75	1001	533	(3.08%)
res7	44	62	18	1.46	1.77	1619	1177	(1.93%)
res8	50	79	19	2.68	3.11	1864	1642	(11.00%)
res9	50	76	23	2.08	2.70	2613	997	(0.04%)
stein1	50	63	20	1.56	1.94	1972	1114	29.94
stein2	50	100	17	2.45	2.90	1801	1282	(0.65%)
stein3	75	94	32	12.75	14.22	5555	3878	(1.03%)
stein4	75	150	33	33.03	35.49	6090	6114	–
n45	45	63	20	1.59	1.98	2072	1112	(8.85%)
n49	45	57	14	0.87	1.04	1363	8560	7.21
n147	147	265	37	235.09	240.75	13640	19114	442.36
g200	200	914	31	2668.21	2679.64	37116	50180	*mem*
t3-0	250	444	52	441.09	446.07	109092	23544	*mem*
t3-1	250	456	53	676.98	681.31	21100	43133	*mem*
t3-2	250	469	42	649.31	654.12	13977	28549	*mem*
t3-3	250	446	50	562.91	564.54	16687	16974	*mem*
t3-4	250	465	39	236.97	247.25	11868	11687	*mem*
t4-0	304	453	55	2782.64	2822.95	18353	23489	*mem*
t4-1	304	457	63	3578.18	3606.15	23796	23250	*mem*
t4-2	304	454	58	3393.84	3437.39	19986	26110	*mem*
t4-3	304	458	67	6824.91	6855.39	29118	39378	*mem*
t4-4	304	468	63	5492.71	5520.48	24549	34754	*mem*

are generated.

The improved performance of the BPC for this uncertainty model cannot be related to a lower-dimensional demand polyhedron as might be between *Asym-G* and *Sym-G* just as we have discussed above, since there is a large increase in the number of variables in *Rob-G* with respect to *Asym-G*. Therefore, we cannot say that it is just the number of variables that affects the performance of the BPC algorithm. Our partial explanation for this is the favourable structure of the polyhedron that results from the Bertsimas and Sim constraints. More specifically, although $2|Q| + 1 = |W|(|W| - 1) + 1$ constraints describe the latter and only $2|W|$ are needed for *Asym-G* (W being the set of terminal nodes and Q the set of pairs of terminals), most of the *Rob-G* constraints are upper- and lower-bounding constraints, perhaps simpler to handle by the LP solver.

3.6 Concluding remarks

We have addressed the network design problem under traffic uncertainty arising when provisioning Virtual Private Networks. Sufficient capacity must be reserved on the links of a large underlying public network to support all valid traffic matrices between a given set of terminals while minimizing total reservation costs.

The first contribution of this study consists of new compact linear MIP formulations which allow us to solve to optimality medium-to-large-size instances of *Asym-G* and *Sym-G* in less than 15 minutes of computing time. These flow-based formulations also provide significant experimental support to the conjecture about *Sym-G* and *Sym-T*, which states that *Sym-G* always has an optimal tree solution.

To exploit traffic statistics that are available for provisioned VPNs, we have investigated a robust VPN provisioning problem, *Rob-G*, which leads to less conservative capacity configurations. The compact linear MIP formulation for this NP-hard problem is more challenging to tackle with a commercial solver than the one for *Asym-G*.

The combined branch-and-price and cutting plane algorithm we have devised, based on path and cut formulations, performs well on large *Sym-G* instances and remarkably well on all *Rob-G* instances. Further work is needed to establish whether it can also compare favourably with the compact formulations for large *Asym-G* instances.

The linear MIP formulations and algorithm presented in this chapter can be extended to the general multi-commodity network design problem under demand uncertainty, where the demand matrix is assumed to lie in a union of polyhedra ([2]).

Chapter 4

Optimal Oblivious OSPF Routing

4.1 Introduction

The importance of effective traffic engineering for highly information dependent economy should not be underestimated. Hence, configuring a vigorous routing strategy to achieve a high customer satisfaction and using network resources efficiently is crucial. Different routing protocols like Multi-Protocol Label Switching (MPLS), Open Shortest Path First (OSPF), Border Gateway Protocol (BGP) etc. can be used to tell routers the best paths to use whereas miscellaneous criteria can be used to determine these paths.

In this chapter, we will consider OSPF routing with a 'fairness' criterion, which is about optimizing network utilization through a 'fair' allocation of the traffic load among the links of the available shortest paths. Particularly, we look for a general OSPF routing strategy enhanced with traffic engineering tools, which is fair for a set of traffic demands, i.e., an optimal oblivious OSPF routing scheme.

The rest of the chapter is organized as follows. In Section 4.2, we make some basic definitions and explain the performance measure we will use in our models to assess the goodness of different routings. Then in Section 4.3, we present our integer programming models for the oblivious routing with general demand

uncertainty. Consequently, we show how we incorporate OSPF routing into our models in Section 4.4. Section 4.5 discusses our Branch-and-Price algorithm while numerical results are provided in Section 4.6. Finally, we offer conclusions in Section 4.7.

4.2 Basic definitions and measures of performance

Unlike the VPN design problem we have discussed in Chapter 3, we consider the general case in this part of the thesis, where any node of the underlying graph G can be the source or the destination of some demand, unless we specify otherwise. Hence, we will slightly modify the notation, which we will briefly review here for the sake of completeness. Consider the undirected graph $G = (V, E)$. All edges $\{h, k\} \in E$ are also referred to as *links*. For each link we have the associated directed pairs (h, k) and (k, h), which we call the *arcs* of G. We denote this set of directed node pairs by A. Moreover, we suppose that each link $\{h, k\}$ is assigned C_{hk} units of capacity, which is available for the total flow on $\{h, k\}$ in both directions. The estimated traffic flow from the source node $s \in V$ to the sink node $t \in V$ is d_{st} where we define the set of such directed source-sink pairs as $Q = \{(s, t) \in V \times V : s \neq t\}$. The traffic matrix (TM) $d = [d_{st}]_{(s,t)\in Q}$ shows the amounts of traffic flow between all directed source-sink pairs.

In accordance with our previous definitions, we denote the fraction of d_{st} routed on the arc (h, k) by f^{st}_{hk}. Then the matrix $f = [f^{st}_{hk}]_{(h,k)\in A,(s,t)\in Q}$ defines a *routing* if it satisfies the following conditions:

$$\sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{4.1}$$

$$0 \leq f^{st}_{hk} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{4.2}$$

and we denote the set of all possible routings on G as Λ. Consequently, the traffic load assigned by $f \in \Lambda$ to the undirected link $\{h, k\} \in E$ for the traffic matrix d is

$L_d^f(hk) = \sum_{(s,t)\in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})$ whereas its utilization is $U_d^f(hk) = L_d^f(hk)/C_{hk}$. The fairness of a routing, i.e., the measure of how balanced the load distribution for a traffic demand d is, can be measured by the maximum link utilization of f ($MaxU_d^f$), that is:

$$MaxU_d^f = \max_{\{h,k\}\in E} U_d^f(hk).$$

Then, the problem of finding the routing with the minimum $MaxU_d^f$ for a fixed TM d is

$$\min_{f\in\Lambda} \{MaxU_d^f\}$$

and it can be modeled as follows:

$$\min r \tag{4.3}$$

$$\text{s.t.} \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{4.4}$$

$$r \geq \sum_{(s,t)\in Q} \frac{d_{st}(f_{hk}^{st} + f_{kh}^{st})}{C_{hk}} \qquad \forall \{h,k\} \in E \tag{4.5}$$

$$\sum_{(s,t)\in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) \leq C_{hk} \qquad \forall \{h,k\} \in E \tag{4.6}$$

$$0 \leq f_{hk}^{st} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q \tag{4.7}$$

where (4.4) ensures that f is a routing and (4.5)-(4.6) imply the existence of a flow, which routes the traffic matrix d respecting the capacity limitations. Notice that (4.5) and (4.6) together with the objective of minimizing r imply that $r \leq 1$, i.e., the traffic load of each link must be less than its capacity. Therefore, (4.6) imposes that no link be overloaded.

4.3 Oblivious routing under polyhedral demand uncertainty

The optimal oblivious routing problem consists in finding a routing for each source-sink pair $(s,t) \in Q$ independent of the traffic matrix d such that the

maximum edge utilization is minimized. In this case, we have a set of traffic matrices D and the best routing is required to support any feasible traffic matrix $d \in D$ in the most balanced way. Thus, oblivious routing yields a conservative strategy with a worst case approach when the demand is uncertain. As a result, the 'goodness' of a routing is assessed based on a set of matrices where the maximum link utilization of a routing f is the highest ratio it achieves over D, i.e., $\max_{d\in D} MaxU_d^f$. However, a more common approach is to use a measure of how close each f is to optimality for any traffic matrix $d \in D$ (Azar et al. [7], Applegate and Cohen [3], Belotti and Pınar [11]). Then the *oblivious ratio* of f on the set D is

$$OR_D^f = \max_{d\in D} \frac{MaxU_d^f}{BEST_d}$$

where $BEST_d$ is the smallest maximum link utilization ratio for d and is equal to the optimal solution of the linear problem (4.3)-(4.7). As a result, the problem of finding the routing with the smallest maximum link utilization for the set D of traffic demands becomes

$$\min_{f\in\Lambda} \max_{d\in D} \frac{\max_{\{h,k\}\in E} U_d^f(hk)}{BEST_d}. \tag{4.8}$$

The problem in (4.8) is mainly a min-max regret problem. The regret for each routing f and traffic matrix d is measured as the ratio of $MaxU_d^f$ to $BEST_d$. As a result, the optimal oblivious routing minimizes the maximum regret over the set D. Notice that $BEST_d$ does not depend on $\{h,k\}$ and hence $\frac{\max_{\{h,k\}\in E} U_d^f(hk)}{BEST_d}$ can be written as $\max_{\{h,k\}\in E} \frac{U_d^f(hk)}{BEST_d}$. Then, we can swap the two max functions in (4.8) to have the equivalent expression

$$\min_{f\in\Lambda} \max_{\{h,k\}\in E} \max_{d\in D} \frac{U_d^f(hk)}{BEST_d}. \tag{4.9}$$

In the sequel, we can model (4.9) as the following mathematical model (OB_{sem}):

$$\min r \tag{4.10}$$

$$\text{s.t.} \quad (4.1), (4.2)$$

$$r \geq \max_{d\in D} \frac{\sum_{(s,t)\in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st})/C_{hk}}{BEST_d} \quad \forall\{h,k\} \in E \tag{4.11}$$

where (4.11) implies that for each link $\{h,k\} \in E$ and routing $f \in \Lambda$, we have a maximization problem over D. Hence the definition of D is important in modeling and solving OB_{sem}.

Unlike the case with fixed traffic demands, although here d is not known it should not be considered as a variable of the optimization model OB_{sem}. It is instead a variable of the inner optimization model on the right-hand side of constraint (4.11). Due to the max operator in constraint (4.11), the model OB_{sem} is equivalent to a semi-infinite optimization model with one constraint (4.11) for each $d \in D$.

Another remark is useful here. In recent works on network design with uncertainty in the traffic demand, there has been an interest towards the set $D' \subseteq D$ of so-called *dominant* demands (see Oriolo [55]), which are defined as those that suffice to describe the entire uncertainty set, or in other words, such that routing all demands in D' implies that all demands in D are also routable. For instance, in network design problems where capacity has to be installed to accommodate a set of uncertain traffic demands, it is easy to prove that a demand d dominates all d' such that $d' \leq d$. A necessary and sufficient condition for dominance between traffic demands has been given by Oriolo [55]. However, the same does not apply here because the objective function of the inner optimization problem is not linear with respect to d, hence for two demands d and d' such that $d' \leq d$, we cannot prove that $\frac{MaxU^f_{d'}}{BEST_{d'}} \leq \frac{MaxU^f_d}{BEST_d}$.

Bearing in mind that the demand uncertainty can be modeled in various ways, we will consider the case of polyhedral uncertainty, i.e., the general traffic uncertainty model

$$D = \{d = [d_{st}]_{(s,t)\in Q} : Ad \leq a,\ d \geq 0\}$$

where $A \in \mathbb{R}^{m\times|Q|}$ and $a \in \mathbb{R}^m$ with m being the number of linear inequalities that define D.

Next, we prove that the semi-infinite optimization model OB_{sem} reduces to its equivalent linear counterpart after a duality transformation similar to the one in Proposition 3.3.1 and Soyster [61].

Firstly, notice that we can write (4.11) as

$$\max_{d\in D}\left\{\sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) - rC_{hk}BEST_d\right\} \leq 0 \quad \forall\{h,k\} \in E. \quad (4.12)$$

Then the left-hand side of (4.12) is a maximization problem, which can be modeled as the following linear programming model (P_{hk}), for each $\{h,k\} \in E$:

$$\max \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) - r\omega C_{hk} \tag{4.13}$$

$$\text{s.t.} \quad \sum_{j:\{s,j\}\in E} (g^{st}_{sj} - g^{st}_{js}) = d_{st} \qquad \forall (s,t) \in Q \tag{4.14}$$

$$\sum_{j:\{i,j\}\in E} (g^{st}_{ij} - g^{st}_{ji}) = 0 \qquad \forall i \in V \setminus \{s,t\}, (s,t) \in Q \tag{4.15}$$

$$\sum_{(s,t)\in Q} (g^{st}_{ij} + g^{st}_{ji}) \leq \omega C_{ij} \qquad \forall \{i,j\} \in E \tag{4.16}$$

$$\omega \leq 1 \tag{4.17}$$

$$\sum_{(s,t)\in Q} a^{st}_z d_{st} \leq a_z \qquad \forall z = 1, \ldots, m \tag{4.18}$$

$$g^{st}_{ij} \geq 0 \qquad \forall (i,j) \in A, (s,t) \in Q \tag{4.19}$$

$$d_{st} \geq 0 \qquad \forall (s,t) \in Q \tag{4.20}$$

$$\omega \geq 0 \tag{4.21}$$

where $\omega = BEST_d$ and the traffic polytope D is defined by m linear inequalities of the form (4.18) such that a_z is the right-hand side of the z^{th} inequality whereas a^{st}_z is the coefficient of d_{st} in that inequality. Applegate and Cohen [3] assume that, at the optimum of the inner optimization problem (4.11), $BEST_d = 1$, i.e., at least one of the arcs is used to its full capacity in the worst case. However, as Belotti and Pınar [11] show, this is not a valid assumption all the time. They give an example of the case where $D = [d_{st}]_{(s,t)\in Q}$ is such that $d_{st} \leq \alpha \frac{\min_{\{h,k\}\in E} C_{hk}}{|Q|} \quad \forall (s,t) \in Q$ with $\alpha < 1$. Then, none of the links would be used totally even if all demands were routed on the link with the minimum capacity. Hence, we avoid such an assumption and use (4.13), (4.16), and (4.17) to model this feature of the problem. Moreover, constraints (4.14)-(4.17) ensure that there is a feasible flow g on $G = (V, E)$ that routes any feasible demand d without violating the link capacities.

For a given r and routing f, P_{hk} is a linear programming problem, and hence we can employ duality to get its dual problem (DP_{hk}) for each link $\{h,k\} \in E$. Consider the dual variables π^{st}_{hk}, $\sigma^{st}_{i,hk}$, $\eta_{ij,hk}$, χ_{hk}, and λ^{hk}_z of the constraints (4.14)

- (4.18), respectively. If we let

$$\Pi^{st}_{i,hk} = \begin{cases} \pi^{st}_{hk} & \text{if } i = s \\ 0 & \text{if } i = t \\ \sigma^{st}_{i,hk} & \text{otherwise} \end{cases} \qquad \forall i \in V, (s,t) \in Q,$$

then for each edge $\{h,k\} \in E$, we have the dual problem DP_{hk}:

$$\min \chi_{hk} + \sum_{z=1}^{m} a_z \lambda^{hk}_z \tag{4.22}$$

$$\text{s.t.} \qquad \Pi^{st}_{i,hk} - \Pi^{st}_{j,hk} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q \tag{4.23}$$

$$-\pi^{st}_{hk} + \sum_{z=1}^{m} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall (s,t) \in Q \tag{4.24}$$

$$-\sum_{\{i,j\} \in E} C_{ij} \eta_{ij,hk} + \chi_{hk} \geq -r C_{hk} \tag{4.25}$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\} \in E \tag{4.26}$$

$$\chi_{hk} \geq 0 \tag{4.27}$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1, ..., m. \tag{4.28}$$

In what follows, we use DP_{hk} and the duality theorems to reduce (4.11) to an equivalent set of linear inequalities.

Proposition 4.3.1. *For the polyhedral traffic uncertainty model where* $D = \{d = [d_{st}]_{(s,t)\in Q} : Ad \leq a,\ d \geq 0\}$ *the right-hand side of the constraint (4.11) for each* $\{h,k\} \in E$ *can be replaced with the equivalent inequality system (4.23)-(4.28) and the inequality*

$$-\chi_{hk} - \sum_{z=1}^{m} a_z \lambda^{hk}_z \geq 0. \tag{4.29}$$

Proof. Suppose D is subject to polyhedral uncertainty. For each link $\{h,k\} \in E$ consider the following LP problem (FP_{hk}) :

$$\{\min 0 : (4.23) - (4.29)\}.$$

Let g^{st}_{ij}, d_{st}, ω, and ψ_{hk} be the dual variables associated with constraints (4.23)-(4.25), and (4.29), respectively. Consequently, the corresponding dual problem (FD_{hk}) is

$$\max\{-\omega r C_{hk} + \sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh})\}$$

$$\text{s.t.} \quad \sum_{j:\{s,j\}\in E} (g^{st}_{sj} - g^{st}_{js}) = d_{st} \qquad \forall (s,t) \in Q$$

$$\sum_{j:\{i,j\}\in E} (g^{st}_{ij} - g^{st}_{ji}) = 0 \qquad \forall i \in V \backslash \{s,t\}, (s,t) \in Q$$

$$\sum_{(s,t)\in Q} (g^{st}_{ij} + g^{st}_{ji}) \leq \omega C_{ij} \qquad \forall \{i,j\} \in E$$

$$\omega \leq \psi_{hk}$$

$$\sum_{(s,t)\in Q} d_{st} a^{st}_z \leq \psi_{hk} a_z \qquad \forall z = 1, ..., m$$

$$(4.19) - (4.21)$$

$$\psi_{hk} \geq 0 \qquad \forall \{h,k\} \in E.$$

Without loss of generality, we can assume that $\psi_{hk} > 0$ since we would have the trivial solution otherwise. Hence, if we scale each variable by ψ_{hk} such that $\tilde{g}^{st}_{ij} = g^{st}_{ij}/\psi_{hk}$, $\tilde{d}_{st} = d_{st}/\psi_{hk}$, and $\tilde{\omega} = \omega/\psi_{hk}$, then FD_{hk} reduces to P_{hk}, i.e., (4.13)-(4.21), as we wanted to show. □

Corollary 4.3.1. *Assuming that the traffic demand set D is subject to polyhedral uncertainty, solving the following LP (OB_{POL}) yields the optimal oblivious routing on $G = (V, E)$:*

$$\min r$$

$$s.t. \quad \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & otherwise \end{cases} \qquad \forall h \in V, (s,t) \in Q$$

$$\chi_{hk} + \sum_{z=1}^{m} a_z \lambda^{hk}_z \leq 0 \qquad \forall \{h,k\} \in E$$

$$\Pi^{st}_{i,hk} - \Pi^{st}_{j,hk} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q, \{h,k\} \in E$$

$$-\pi^{st}_{hk} + \sum_{z=1}^{m} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \qquad \forall (s,t) \in Q,\ \{h,k\} \in E$$

$$-\sum_{\{i,j\}\in E} C_{ij}\eta_{ij,hk} + \chi_{hk} \geq -rC_{hk} \qquad \forall \{h,k\} \in E$$

$$0 \leq f^{st}_{hk} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\}, \{h,k\} \in E$$

$$\chi_{hk} \geq 0 \qquad \forall \{h,k\} \in E$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1, \ldots, m,\ \{h,k\} \in E.$$

An important issue that we need to highlight here is that OB_{POL} is a linear programming problem. Hence, we proved that the optimal oblivious ratio for unconstrained routing[1] under general traffic uncertainty can be computed in polynomial time by solving OB_{POL}.

Before discussing how the OSPF routing protocol could be incorporated into the oblivious routing problem, it would be useful to mention that there is no condition on how the routes would be chosen to carry the feasible traffic matrices on $G = (V, E)$ in the above model. Moreover, the fractional flow variables imply that the flow from s to t can be split in any fraction among the paths defined between them. However, this latter issue is not applicable with the current traffic engineering technology. The current approach is either to use a unique path

[1]With a slight abuse of notation we will call the unconstrained routing as MPLS in the rest of this thesis. We do that since the routing paths are not dependent on any link metric in MPLS routing. However, we should note that MPLS can also be combined with constraint-based routing so as to compute the set of routes to satisfy some Quality-of-Service requirements imposed by the administrative policies.

for each demand d_{st} or to split it *equally* among multiple paths. Besides, OSPF protocol is compatible with the current applications. Hence the inclusion of OSPF routing with the aim of 'fair' traffic allocation is also important for the sake of applicability. This is discussed in the following section.

4.4 Modeling OSPF routing

Open Shortest Path First (OSPF) routing protocols route flow between node pairs along the corresponding shortest paths defined with respect to some metric. The traditional approach is to fix this metric in advance and determine the shortest paths a priori. The more recent approach is to manage the OSPF metric so as to optimize a given design criteria. We also adopt this latter approach since we believe in the necessity of good weight management to improve the OSPF performance. Therefore, we consider OSPF routing with the goal of minimizing the oblivious ratio r defined in (4.8), where $BEST_d$ is the oblivious ratio of the best non-OSPF routing, since we want to compare the performance of OSPF with that of the best non-constrained routing.

Naturally, there can be more than one shortest path between a pair of nodes. In terms of routing design, one option is to consider unsplittable routing such that each demand d_{st} can be routed on a unique path. However, using multiple paths would mostly improve the fairness of work load distribution. To further clarify this issue, consider the simple example given in Figure 4.1. The numbers on each link are its weight and capacity, respectively. For example, the link $\{A, B\}$ is assigned a unit weight and 12 units of capacity, which is available for the traffic in both directions.

Suppose that we have a fixed traffic matrix d where nodes A and C exchange some traffic with $d_{AC} = 8$ and $d_{CA} = 4$. In this case, we can define 3 shortest paths between A and C in both directions. With unsplittable routing we would route each demand along a single path. Then we may have the situation shown in Figure 4.1a where the link $\{A, C\}$ is used to its full capacity although the rest

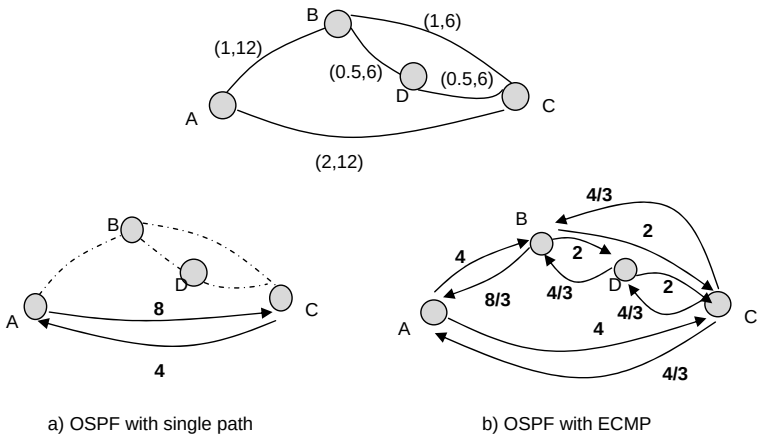

Figure 4.1: Example for splittable vs unsplittable routing.

of the links are left idle. On the other hand, if we allow splittable routing, then we would have the case in Figure 4.1b where the utilization of all links are around 50%. This is important since it helps to improve the availability and reliability of the network. Naturally, those links with high utilization rates would fail earlier and more frequently than the other network links, which is counterproductive for network availability. Hence rather than unsplittable routing, it would be better to apply Equal Cost Multi-Path (ECMP) routing in which the demand d_{st} accumulated at some node h is split evenly among all shortest paths between h and t. As a result, we model OSPF routing with ECMP. Two formulations, namely the flow formulation and the tree formulation, will be presented in the rest of this section.

4.4.1 Variables and parameters

Below we present two formulations to model OSPF routing. Integer variables θ_{hk} define the metric used at each arc (h, k) and range between 1 and Θ_{max}, which is a parameter of value 65535[2]. Although we are only interested in the optimal values of the f and θ variables, some auxiliary classes of variables are needed. We define ρ_h^t as the shortest path distance between h and t according to the metric defined by the θ_{hk} variables. In order to impose ECMP constraints, we use a

[2]This is the common constant used in the literature when integer link weights are required.

variable φ_h^{st} which gives the fraction of flow that, after entering node h, is split among different outgoing arcs due to the ECMP rules. For example, in Figure 4.1b, for node B and the demand from A to C we have $\varphi_B^{AC} = 0.25$ because the portion of flow from A to B, i.e., 50% of d_{AC}, is equally split on the two shortest paths from B to C. Similarly, $\varphi_C^{CA} = 1/3$ since there are three shortest paths from C to A.

4.4.2 Flow formulation

To model OSPF routing, we must ensure that the demands are routed on the corresponding shortest paths. We can do so via a set of linear inequalities where the binary variable y_{hk}^t indicates if the arc (h, k) is on some shortest path destined to node t, i.e., if it is a shortest path arc for t. Note that OSPF is a source invariant routing scheme, and this is the reason why we do not need an index for the source node s in y variables. Consequently, we use the constraints

$$f_{hk}^{st} \leq y_{hk}^t \quad \forall (h,k) \in A, (s,t) \in Q \tag{4.30}$$

to relate y variables to flow variables. Moreover we include

$$y_{hk}^t + \rho_k^t - \rho_h^t + \theta_{hk} \geq 1 \qquad \forall (h,k) \in A, t \in V \tag{4.31}$$

$$-y_{hk}^t - \frac{\rho_k^t - \rho_h^t + \theta_{hk}}{2\Theta_{max}} \geq -1 \quad \forall (h,k) \in A, t \in V \tag{4.32}$$

to model OSPF routing. The Bellman conditions, $\rho_k^t - \rho_h^t + \theta_{hk} \geq 0$, imposing non-negative reduced cost of arc (h, k) for the set of shortest paths destined to t, are dominated by constraints (4.31), and therefore are not included. If $y_{hk}^t = 1$, then (h, k) is a shortest path arc for all demands destined to t and hence the Bellman condition must be satisfied with equality, as imposed by (4.31) and (4.32).

On the other hand, if some arc (h, k) is not a shortest path arc to t according to weights θ, then its reduced cost must be at least 1 since we require $\theta_{hk} \geq 1$. Lastly, any constant larger than $2\Theta_{max}$ can be used in (4.32). Since for each link $\{h, k\} \in E$, we have $(h, k) \in A$ and $(k, h) \in A$, we use $2\Theta_{max}$ as mentioned in Holmberg and Yuan [32]. Although they explain the reason for such a choice,

we restate it here for the sake of completeness. Firstly, for some arc (k, h) we know $\rho_h^t \geq \rho_k^t - \theta_{kh}$ by the Bellman conditions. As a result, for arc (h, k), we have $\rho_k^t - \rho_h^t + \theta_{hk} \leq \rho_k^t - \rho_k^t + \theta_{kh} + \theta_{hk} \leq 2\Theta_{max}$. Finally, we need the following set of constraints since we want to apply the ECMP rule to implement splittable routing:

$$f_{hk}^{st} \leq \varphi_h^{st} \qquad \forall (h,k) \in A, (s,t) \in Q \tag{4.33}$$

$$1 + f_{hk}^{st} - \varphi_h^{st} \geq y_{hk}^t \quad \forall (h,k) \in A, (s,t) \in Q \tag{4.34}$$

with the variable bounds

$$\begin{aligned} 1 \leq \theta_{hk} \leq \Theta_{max} \quad & \text{integer } \forall (h,k) \in A \\ y_{hk}^t \in \{0,1\} \quad & \forall (h,k) \in A, t \in V \\ 0 \leq \varphi_h^{st} \leq 1 \quad & \forall h \in V, (s,t) \in Q. \end{aligned}$$

Constraints (4.33) and (4.34) impose that if demand d_{st} is routed via some node h, then all arcs originating at h and contained in some shortest path to t should share the total flow accumulated in h equally.

Corollary 4.4.1. *The solution of the following linear MIP ($FLOW_{OSPF}$) is the optimal oblivious OSPF routing on $G = (V, E)$ with equal load sharing under polyhedral demand uncertainty:*

$$\min r$$

$$s.t. \quad \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & otherwise \end{cases} \qquad \forall h \in V, (s,t) \in Q$$

$$\chi_{hk} + \sum_{z=1}^{m} a_z \lambda_z^{hk} \leq 0 \qquad \forall \{h,k\} \in E$$

$$\Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} \geq 0 \qquad \forall (i,j) \in A,\ (s,t) \in Q,\ \{h,k\} \in E$$

$$-\pi_{hk}^{st} + \sum_{z=1}^{m} a_z^{st} \lambda_z^{hk} \geq f_{hk}^{st} + f_{kh}^{st} \qquad \forall (s,t) \in Q,\ \{h,k\} \in E$$

$$-\sum_{\{h,k\}\in E} C_{ij}\eta_{ij,hk} + \chi_{hk} \geq -rC_{hk} \qquad \forall \{h,k\} \in E$$

$$f_{hk}^{st} \leq y_{hk}^{t} \qquad \forall (h,k) \in A, (s,t) \in Q$$

$$y_{hk}^{t} + \rho_k^t - \rho_h^t + \theta_{hk} \geq 1 \qquad \forall (h,k) \in A, t \in V$$

$$-y_{hk}^{t} - \frac{\rho_k^t - \rho_h^t + \theta_{hk}}{2\Theta_{max}} \geq -1 \qquad \forall (h,k) \in A, t \in V$$

$$f_{hk}^{st} \leq \varphi_h^{st} \qquad \forall (h,k) \in A, (s,t) \in Q$$

$$1 + f_{hk}^{st} - \varphi_h^{st} \geq y_{hk}^{t} \qquad \forall (h,k) \in A, (s,t) \in Q$$

$$0 \leq f_{hk}^{st} \leq 1 \qquad \forall (h,k) \in A, (s,t) \in Q$$

$$\eta_{ij,hk} \geq 0 \qquad \forall \{i,j\}, \{h,k\} \in E$$

$$\chi_{hk} \geq 0 \qquad \forall \{h,k\} \in E$$

$$\lambda_z^{hk} \geq 0 \qquad \forall z = 1, \ldots, m,\ \{h,k\} \in E$$

$$1 \leq \theta_{hk} \leq \Theta_{max} \qquad integer\ \forall (h,k) \in A$$

$$y_{hk}^{t} \in \{0,1\} \qquad \forall (h,k) \in A, t \in V$$

$$0 \leq \varphi_h^{st} \leq 1 \qquad \forall h \in V, (s,t) \in Q.$$

Notice that if we use the flow formulation $FLOW_{OSPF}$ to model OSPF with ECMP, we need $2|E|(|V|+1) + |V|^3$ additional variables, $2|E||V|$ of which are binary. Even for medium sized networks, the size of our formulation can get very large and the related solution time would be too long. Hence, it can be quite time consuming to solve these problems using off-the-shelf MIP solvers. Therefore, we propose an alternative tree formulation, whose linear relaxation can be solved by

column generation, in the next section.

4.4.3 Alternative formulation

In this section we adopt an alternative approach where we use tree rather than flow variables to model OSPF routing. In this model each tree variable corresponds to an *SP tree*, which is a widely used structure in OSPF and IS-IS routing protocols. We give a brief explanation of these special structures in the following lines and later proceed with a discussion of our tree formulation in the rest of this section.

4.4.3.1 Shortest Paths trees

A Shortest Paths tree (*SP tree*) is an acyclic graph such that, for at least one metric, all and only paths within the *SP tree* are the shortest ones. In other words, an *SP tree* T with respect to some node t (destination node) of the backbone graph $G = (V, E)$ contains all shortest paths from all other nodes of G to t for a given vector of link weights. Notice that if there are multiple shortest paths from some node $s \in V \setminus \{t\}$ to t, then all of them are included in T. Hence, the structure of an *SP tree*, i.e., the set of arcs it contains, is very much affected by the link metric. Therefore we need to underline that an *SP tree* T does not have to be a tree literally since it is the union of some paths. As a result, an *SP tree* is some acyclic subgraph of G, but not a tree in general.

4.4.3.2 Tree formulation

In this second formulation ($TREE_{OSPF}$) of our problem, we use destination based *SP trees* and hence T shows how all the traffic towards its destination node t should be routed on the arcs of the backbone graph. In other words, each *SP tree* T defines a routing configuration for its root node. Thus, we want only one *SP tree* to be used for each $t \in V$. We model this requirement via binary τ_T^t variables,

which indicate whether the implicitly defined *SP tree* T is used to route all traffic flow ending at t or not. Bearing in mind that the number of paths in a graph can be exponential in number, we define Ω_t as the set of *SP* trees with destination t and Ω_{hk} as the set of *SP* trees containing arc (h,k). Consequently, we can ensure that a single *SP tree* would be used for each destination by the constraint

$$\sum_{T\in\Omega_t} \tau_T^t = 1 \quad \forall t \in V. \tag{4.35}$$

Moreover, we use the inequality

$$f_{hk}^{st} \leq \sum_{T\in\Omega_t\cap\Omega_{hk}} \tau_T^t \quad \forall (h,k)\in A, (s,t)\in Q \tag{4.36}$$

to relate the τ variables to flow variables. Notice that (4.36) is analogous to (4.30) of the flow formulation. If some flow originated at s and destined to t is routed on arc (h,k), then this arc should be a shortest path arc for t in any *SP tree* T that will be used for it. Hence the sum on the right-hand side of (4.36) must be 1, which ensures that the *SP tree* for t contains (h,k).

Consequently, we include the OSPF constraints

$$\sum_{T\in\Omega_t\cap\Omega_{hk}} \tau_T^t + \rho_k^t - \rho_h^t + \theta_{hk} \geq 1 \qquad \forall t\in V, (h,k)\in A \tag{4.37}$$

$$-\sum_{T\in\Omega_t\cap\Omega_{hk}} \tau_T^t - \frac{\rho_k^t - \rho_h^t + \theta_{hk}}{2\Theta_{max}} \geq -1 \quad \forall t\in V, (h,k)\in A, \tag{4.38}$$

which are similar to (4.31) and (4.32), respectively. Note that the summations in (4.37) and (4.38) would be equal to one only for the shortest path arcs ensuring that their reduced costs are zero. The final set of constraints are the following ECMP constraints

$$f_{hk}^{st} \leq \varphi_h^{st} \qquad \forall (h,k)\in A, (s,t)\in Q \tag{4.39}$$

$$1 + f_{hk}^{st} - \varphi_h^{st} \geq \sum_{T\in\Omega_t\cap\Omega_{hk}} \tau_T^t \quad \forall (h,k)\in A, (s,t)\in Q \tag{4.40}$$

with the variable bounds

$$\begin{aligned} 1 \leq \theta_{hk} \leq \Theta_{max} &\quad \text{integer} \quad \forall (h,k)\in A \\ \tau_T^t \in \{0,1\} &\quad \forall t\in V, T\in\Omega_t \\ 0 \leq \varphi_h^{st} \leq 1 &\quad \forall h\in V, (s,t)\in Q. \end{aligned}$$

The flow and tree formulations are analogous to each other, and the difference is how one tries to solve them. Before discussing our solution approach for the tree formulation, we should make a remark here. As we have mentioned before, the *SP trees* are defined by the weight metric θ, which is also a variable of our model. Hence, we know neither the number nor the structure of *SP trees* explicitly in advance, and we can say the sets Ω_t and Ω_{hk} are implicitly defined for every destination node t and network arc $\{h, k\}$.

Consequently, the oblivious routing model discussed in Proposition 4.3.1 can be combined with one of the $FLOW_{OSPF}$ or $PATH_{OSPF}$ models to find the optimal OSPF routing under ECMP rule such that the oblivious ratio is minimum.

4.5 A Branch-and-Price algorithm for exact solution

The number of paths in a graph depends on the structure of the graph, and it can be huge. So can be the number of variables in the tree formulation. Hence, we have decided to develop a branch-and-price (B&P) algorithm, which is a column generation integrated branch-and-bound technique. This method was initially discussed in Barnhart et al. [10] and it is an efficient approach to cope with those models with a large number of variables. Basically, it is a modified branch-and-bound (B&B) algorithm, which starts with a restricted LP relaxation (RLP_0) with fewer variables than the original problem and applies column generation at each node of the B&B tree. The subproblem in a B&B node (RLP_{curr}) is optimal when no new columns can be added to the problem and branching occurs if the integrality conditions are not satisfied by the current solution. We have already discussed an application of the B&P algorithm strengthened by including cutting planes in the VPN design problem in Section 3.4.

In our problem, we consider destination based *SP trees* comprising shortest paths to each node $t \in V$ from all other nodes of the graph $G = (V, E)$. Just like the number of paths in a graph, the number of *SP trees* can also be very large.

Therefore, it is wise to use B&P to solve the tree formulation. We summarize the main steps of our B&P algorithm in Algorithm 4.1. The details of the application are addressed in the rest of this section.

As a final remark, note that we use the terms *SP tree T destined at node t* and τ_T^t *variable* interchangeably throughout this section.

Algorithm 4.1 B&P algorithm

Input: undirected graph $G = (V, E)$, traffic polytope D, link capacity vector $\vec{C}$;
Output: optimal oblivious ratio for OSPF routing and $(G, D, \vec{C})$;

Initialize:
  Find an initial set Ω_0 of *SP trees*, i.e., τ_T^t variables;
  $\tilde{\Omega} \leftarrow \Omega_0$; // $\tilde{\Omega}$: current set of *SP trees*;
  $S \leftarrow \{root\}$; // S : the current set of unevaluated B&P nodes
                // *root* : root node of the B&P tree;
  $UB \leftarrow \infty$;
while $S \neq \emptyset$ **do**
  $n_b = \text{argmin}_{n \in S} LB(n)$;
  $S \leftarrow S \backslash \{n_b\}$;
  repeat
    Optimize: Get $z^*(n_b, \tilde{\Omega})$;//optimal value of RLP_{curr}
    Price:
    for all $t \in V$ **do**
      Search for a new $\tau_{\hat{T}}^t$ variable, i.e., an *SP tree* $\hat{T}$ destined to t;
      if $\tau_{\hat{T}}^t$ has a promising reduced cost **then**
        $\tilde{\Omega} = \tilde{\Omega} \cup \hat{T}$; // update the current set of *SP trees*
        Update RLP_{curr};
  until no new $\hat{T}$ can be found
  if current LP is feasible **then**
    Let $z_{ub}^*(n_b)$ be the upper bound obtained by approximation;
    if $z_{ub}^*(n_b) < UB$ **then**
      $UB \leftarrow z_{ub}^*(n_b)$;
    if the current optimal solution is not integral **then**
      Branch:
        Select a fractional $\bar{\tau}_T^t$ variable and branch;
        Create two child nodes $\{n_r, n_l\}$ and let $S = S \cup \{n_r, n_l\}$;
  Extract B&P nodes that are fathomed by bound or infeasibility from S.

4.5.1 Initialization

We start our B&P algorithm with a relaxed formulation RLP_0, whose solution is feasible but not necessarily optimal for the original problem. Namely, the initial model RLP_0 is as follows:

$$
\begin{aligned}
& \min r \\
\text{s.t.} \quad \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) &= \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} && \forall h \in V, (s,t) \in Q \\
\chi_{hk} + \sum_{z=1}^{m} a_z \lambda_z^{hk} &\leq 0 && \forall \{h,k\} \in E \\
\Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} &\geq 0 && \forall (i,j) \in A, (s,t) \in Q, \{h,k\} \in E \\
-\pi_{hk}^{st} + \sum_{z=1}^{m} a_z^{st} \lambda_z^{hk} &\geq f_{hk}^{st} + f_{kh}^{st} && \forall (s,t) \in Q, \{h,k\} \in E \\
-\sum_{\{i,j\}\in E} C_{ij}\eta_{ij,hk} + \chi_{hk} &\geq -rC_{hk} && \forall \{h,k\} \in E \\
\sum_{T\in\Omega_t^0} \tau_T^t &= 1 && \forall t \in V \qquad (4.41) \\
f_{hk}^{st} &\leq \sum_{T\in\Omega_t^0\cap\Omega_{hk}^0} \tau_T^t && \forall (h,k) \in A, (s,t) \in Q \\
\sum_{T\in\Omega_t^0\cap\Omega_{hk}^0} \tau_T^t + \rho_k^t - \rho_h^t + \theta_{hk} &\geq 1 && \forall t \in V, (h,k) \in A \\
-\sum_{T\in\Omega_t^0\cap\Omega_{hk}^0} \tau_T^t - \frac{\rho_k^t - \rho_h^t + \theta_{hk}}{2\Theta_{max}} &\geq -1 && \forall t \in V, (h,k) \in A \\
f_{hk}^{st} &\leq \varphi_h^{st} && \forall (h,k) \in A, (s,t) \in Q \\
1 + f_{hk}^{st} - \varphi_h^{st} &\geq \sum_{T\in\Omega_t^0\cap\Omega_{hk}^0} \tau_T^t && \forall (h,k) \in A, (s,t) \in Q \\
1 \leq \theta_{hk} &\leq \Theta_{max} && \text{integer } \forall (h,k) \in A \\
\tau_T^t &\in \{0,1\} && \forall t \in V, T \in \Omega_t^0 \\
0 \leq \varphi_h^{st} &\leq 1 && \forall h \in V, (s,t) \in Q
\end{aligned}
$$

where $\Omega_t^0 = \Omega_0 \cap \Omega_t$ and $\Omega_{hk}^0 = \Omega_0 \cap \Omega_{hk}$ with the inital set of *SP trees* Ω_0. Given constraints (4.41), we must have at least one *SP tree* for every node $t \in V$ in Ω_0 to

ensure that each t is reachable from every other node of $G = (V, E)$. Therefore, we need to construct Ω_0 using some metric. This metric can be such that each arc is assigned a unit weight or a value proportional to the physical distance between its two endpoints. We prefer to use an *inverse capacity* weight setting in which the weight of each arc is equal to the inverse of its capacity (this has been used for instance in some network operated by Cisco). This choice implies that Ω_0 will be formed by considering the link capacities to some extent. Finally, notice that $|\Omega_0| = |V|$ and we have one τ_T^t variable for each t in RLP_0. Hence we start with $|V|$ binary variables, which is much less than $2|E||V|$ of the flow formulation.

4.5.2 Pricing

In each node of the B&P tree, the linear programming problem is solved by generating the necessary τ variables dynamically. Given a solution of RLP_{curr} which uses a subset of τ variables, a *pricing* procedure is used to find a set of new τ variables whose reduced cost is negative and which may therefore improve the current routing. In other words, we look for some more promising routing strategies. Notice that the reduced cost of each τ_T^t variable (red_T^t) is

$$-\zeta_t - \sum_{(h,k)\in T} \left[\upsilon_{hk}^t - \varsigma_{hk}^t + \sum_{s\in V\setminus\{t\}} (\nu_{hk}^{st} - \kappa_{hk}^{st}) + \sum_{c\in cut(n_b)} B_c \emptyset_{hk}^c \right], \qquad (4.42)$$

where ζ_t, ν_{hk}^{st}, υ_{hk}^t, ς_{hk}^t, and κ_{hk}^{st} are the dual variables of the constraints (4.35),(4.36), (4.37), (4.38), and (4.40), respectively. Moreover, we take care of the dual variables for the branching rules by including $B_c\emptyset_{hk}^c$ in (4.42). In brief, suppose we are in the B&P node n_b with $cut(n_b)$ being the set of cutting planes added for all ancestor nodes of n_b. Then consider the *SP tree* T. If the arc (h, k) is contained in the *SP tree* T, then we would have $\emptyset_{hk}^c = 1\ \forall c \in cut(n_b)$ provided that T appears in the cutting plane c. As a result, the dual variables B_c of the corresponding branching rules will be included in the reduced cost of τ_T^t.

As we have expressed in Section 4.5.1, we start the B&P algorithm with an initial set Ω_0 of *SP trees*. Then, as we generate new τ variables, we include the corresponding *SP trees* in our model, and update the set of currently available

SP trees ($\tilde{\Omega}$) accordingly. While red_T^t is nonnegative for all *SP trees*, which are enumerated so far, i.e., $\forall\ T \in \tilde{\Omega}$, if we can find a new $\tau_{\hat{T}}^t$ with a negative reduced cost, then we can improve the current solution by simply routing all demands destined to t on $\hat{T}$.

To determine such *SP trees* we solve a shortest path problem for each destination node $t \in V$ with arc metric α on an auxiliary graph $G_\alpha(t)$. Two important issues should be handled with care at this stage. First, the solution of the pricing problem must comply with the definition of an *SP tree*, i.e., ECMP routing and integer arc weights must be ensured. Second, we can guarantee to have neither a nonnegative α nor an acyclic $G_\alpha(t)$. Actually, it is very likely that $G_\alpha(t)$ has negative cycles. Hence we cannot use the well known shortest path algorithms like Djikstra or Bellman-Ford algorithms to solve the pricing problem. Therefore, for each destination node t, we solve the pricing problem to determine promising *SP trees* using the following MIP model (PR_t):

$$z_t^* = \min \sum_{(h,k)\in A} \alpha_{hk} y_{hk}$$

$$\text{s.t.} \quad \sum_{k:\{h,k\}\in E} (f_{hk}^s - f_{kh}^s) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, s \in V\backslash\{t\}$$

$$f_{hk}^s \le \varphi_h^s \qquad \forall (h,k) \in A, s \in V\backslash\{t\}$$

$$1 + f_{hk}^s - \varphi_h^s \ge y_{hk} \qquad \forall (h,k) \in A, s \in V\backslash\{t\}$$

$$-y_{hk} - \left(\frac{\rho_k - \rho_h + \theta_{hk}}{2\Theta_{max}}\right) \ge -1 \qquad \forall (h,k) \in A$$

$$y_{hk} + \rho_k - \rho_h + \theta_{hk} \ge 1 \qquad \forall (h,k) \in A$$

$$0 \le f_{hk}^s \le 1 \qquad \forall (h,k) \in A, s \in V$$

$$0 \le \varphi_h^s \le 1 \qquad \forall h \in V, s \in V$$

$$1 \le \theta_{hk} \le \Theta_{max} \qquad \text{integer } \forall (h,k) \in A$$

$$y_{hk} \in \{0,1\} \qquad \forall (h,k) \in A$$

$$\rho_h \ge 0 \qquad \forall h \in V$$

where the binary variable y_{hk} indicates if (h,k) is an *SP arc* for t whereas f, φ, ρ, and θ retain their definitions made in the original master problem. Moreover, we

set

$$\alpha_{hk} = -\bar{v}^t_{hk} + \bar{\varsigma}^t_{hk} - \sum_{s \in V \setminus \{t\}} (\bar{\nu}^{st}_{hk} - \bar{\kappa}^{st}_{hk}) - \sum_{c \in cut(n_b)} B_c \phi^{n_b}_{hk} \;\; \forall (h,k) \in A$$

in the objective function. Consequently, since PR_t contains the OSPF and the ECMP constraints as we have discussed before, its solution is an *SP tree* $\hat{T}$ defined with respect to some metric θ and its total length is $z^*_t = \sum_{(h,k) \in T^*} \alpha_{hk}$. Now, if $z^*_t < \bar{\zeta}_t$ then we have a new routing configuration whose inclusion could improve the current solution of the original problem. Hence we add the *SP tree* $\hat{T} = \{(h,k) \in A : y^*_{hk} = 1\}$ destined at t to $\tilde{\Omega}$. Note that we solve the pricing problem for all nodes $t \in V$ at each call of the *Price* routine in Algorithm 4.1.

4.5.3 Upper bound approximation

At each node n_b of the B&P tree, we keep on pricing τ variables and reoptimizing the updated RLP_{curr} problem until we cannot identify new *SP trees*. When we are done at n_b, we have a lower bound $LB(n_b)$ on the optimal oblivious ratio $r(n_b)$, we could achieve under the same set of constraints defining n_b. On the other hand, if we can find a feasible solution of the original master problem, then this will be an upper bound (UB) on the optimal oblivious ratio r^*. Such an information would be useful especially for those large instances that are difficult to solve to optimality in reasonable time. As a result, we implement a simple method where we fix an *SP tree* for each destination node $t \in V$ and solve the original problem with this specific routing plan. In brief, given the optimal solution of RLP_{curr} we have the optimal values for the $\bar{\tau}^t_T$ variables. So, for each $t \in V$, we pick the *SP tree* T^* destined at t such that $\bar{\tau}^t_{T^*} \geq \bar{\tau}^t_T \;\; \forall T \in \tilde{\Omega}_t$, where $\tilde{\Omega}_t$ is the set of currently known *SP trees* destined at t. Then we fix these $\bar{\tau}^t_{T^*}$ variables to 1 and solve the original master problem. If this routing strategy is viable, then we have an oblivious ratio $z_{ub}(n_b)$, which is an upper bound UB on the optimal oblivious ratio r^*. An overview of this method is provided in Procedure 4.2.

Procedure 4.2 Upper bound approximation

Input: optimal values of the τ variables for RLP_{curr}, i.e.,$\bar{\tau}$; and the corresponding *SP trees*;

Output: upper bound UB on the optimal oblivious ratio r;

for all $t \in V$ **do**

$T^* = \text{argmax}_{T \in \Omega_t} \tau_T^t$;

Let $\bar{A}_t$ be the set of *SP arcs* contained in T^*;

Get the fraction of demand routed on each arc $(h,k) \in \bar{A}_t$ by solving LP_t:

$$\min 0$$

$$\text{s.t. } \sum_{(h,k)\in\bar{A}_t} f_{hk}^s - \sum_{(k,h)\in\bar{A}_t} f_{kh}^s = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, s \in V \backslash \{t\}$$

$$-f_{hk}^s + \varphi_h^s = 0 \qquad \forall (h,k) \in \bar{A}_t, s \in V \backslash \{t\}$$

$$0 \le f_{hk}^s \le 1 \qquad \forall (h,k) \in \bar{A}_t, s \in V \backslash \{t\}$$

$$f_{hk}^s = 0 \qquad \forall (h,k) \in A \backslash \bar{A}_t, s \in V \backslash \{t\}$$

$$0 \le \varphi_h^s \le 1 \qquad \forall h \in V, s \in V \backslash \{t\};$$

$\bar{\sigma}_{hk}^{st} \leftarrow \bar{f}_{hk}^s + \bar{f}_{kh}^s \quad \forall \{h,k\} \in E, (s,t) \in Q$;

Solve the following problem P_{UB} to get an upper bound UB on r^*:

$$z_{ub}^*(n_b) = \min r$$

$$\text{s.t.} \quad \chi_{hk} + \sum_{z=1}^m a_z \lambda_z^{hk} \le 0 \qquad \forall \{h,k\} \in E$$

$$\Pi_{i,hk}^{st} - \Pi_{j,hk}^{st} + \eta_{ij,hk} \ge 0 \qquad \forall (i,j) \in A, (s,t) \in Q, \{h,k\} \in E$$

$$-\pi_{hk}^{st} + \sum_{z=1}^m a_z^{st} \lambda_z^{hk} \ge \bar{\sigma}_{hk}^{st} \qquad \forall (s,t) \in Q, \{h,k\} \in E$$

$$-\sum_{\{i,j\}\in E} C_{ij} \eta_{ij,hk} + \chi_{hk} \ge -r C_{hk} \qquad \forall \{h,k\} \in E$$

$$\rho_k^t - \rho_h^t + \theta_{hk} = 0 \qquad \forall (h,k) \in \bar{A}_t, t \in V$$

$$\rho_k^t - \rho_h^t + \theta_{hk} \ge 0 \qquad \forall (h,k) \in A \backslash \bar{A}_t, t \in V$$

$$\chi_{hk} \ge 0 \qquad \forall \{h,k\} \in E$$

$$\lambda_z^{hk} \ge 0 \qquad \forall z = 1, .., m, \{h,k\} \in E$$

$$\eta_{ij,hk} \ge 0 \qquad \forall \{i,j\} \in E, \ \{h,k\} \in E$$

$$\rho_h^t \ge 0 \qquad \forall h \in V, \ t \in V$$

$$1 \le \theta_{hk} \le \Theta_{max} \qquad \text{integer } \forall (h,k) \in A$$

$$r \ge 1;$$

if $z_{ub}^*(n_b) < UB$ **then**

$UB \leftarrow z_{ub}^*(n_b)$

4.5.4 Branching

The efficiency of the B&P algorithm is highly dependent on the effectiveness of the branching rule. Moreover, the structure of the pricing problem should not be destroyed for the B&P method to be applicable. Hence, we use a branching rule that exploits the problem structure to partition the solution space without complicating the pricing problem.

As we have mentioned in Algorithm 4.1, we use fractional $\bar{\tau}_T^t$ variables to determine the restrictions we impose in each branching step. This does not mean that we base our branching rule on the dichotomy of these variables. Such an approach would not be efficient since the algorithm might get stuck to the same set of *SP trees* and loop. Suppose that we have used a branching rule such that $\tau_T^t = 0$ in one branch and $\tau_T^t = 1$ in the other. The former condition means that the *SP Tree* T cannot be used for the destination node t. However, it is possible that PR_t finds an *SP Tree* $\tilde{T}$ with exactly the same set of arcs of T, i.e., $\tilde{T} \equiv T$. Consequently, we have decided to create two subdivisions of the current problem based on an arc (h^*, k^*) being or not being an *SP arc* for the demand $d_{s^*t^*}$ of the pair (s^*, t^*). The procedure for selecting the quadruple (h^*, k^*, s^*, t^*) is explained in Procedure 4.3.

When we are done with branch selection, we use the following rule to partition the solution space by creating two new nodes such that either of the following conditions holds:

- (h^*, k^*) is not an *SP arc* for the pair (s^*, t^*), i.e.,

$$f_{h^*k^*}^{s^*t^*} = 0. \tag{4.43}$$

- (h^*, k^*) is an *SP arc* for the pair (s^*, t^*), i.e.,

$$f_{h^*k^*}^{s^*t^*} \geq \frac{\sum_{(k,h^*)\in A} f_{kh^*}^{st}}{deg(h^*) - 1}. \tag{4.44}$$

Notice that the summation on the right hand side of the inequality (4.44) is the total inflow for node h^*. Moreover, suppose $deg(h^*)$ arcs are incident to h^*.

Procedure 4.3 Branch selection.

Input: $\bar{\tau}_T^t$ values in the solution of RLP_{curr}
Output: The quadruple (h^*, k^*, s^*, t^*) // $(s^*, t^*) \in Q$; $(h^*, k^*) \in A$

```
Take the most fractional τ̄^t_{T*}; t* ← t and T_1 ← T*
Find the second most fractional τ̄^{t*}_{T_*}; let T_2 ← T_*
found ← FALSE
for all (h,k) ∈ A do
  if (h,k) ∈ T* ∪ T_* and (h,k) ∉ T* ∩ T_* then
    if f̄^{st}_{hk} > 0 and (h,k,s,t) is not used in upper branches then
      if deg(h) > 1 then
        (h*,k*,s*,t*) ← (h,k,s,t)
        found ← TRUE
if found = FALSE then
  for all (h,k) ∈ A do
    for all (s,t) ∈ Q do
      if f^{st}_{hk} > 0 and (h,k,s,t) is not used in upper branches then
        (h*,k*,s*,t*) ← (h,k,s,t)
        found ← TRUE
if found = FALSE then
  STOP //fathom the current B&P node
```

Then, in order (h^*, k^*) to be an *SP arc*, we must have at least one incoming arc and at most $deg(h^*)-1$ outgoing arcs for node h^*. Hence in the most splitted case, all arcs departing from node h^* would be *SP arcs* and the total flow accumulated in h^* will be splitted evenly among them according to the ECMP routing rule. This is why we have this constant in the denominator of (4.44).

Given the current B&P node n_b and its associated relaxation RLP_{curr}, we create two new nodes n_r and n_l by adding the constraints (4.43) and (4.44) to the current restricted problem as well as the corresponding pricing problems PR_{t^*}. Additionally, we also impose the constraints

- Do not use *SP trees* containing arc (h^*, k^*), i.e.,

$$\sum_{T \in \tilde{\Omega}_{t^*} \cap \tilde{\Omega}_{h^* k^*}} \tau_T^{t^*} = 0. \tag{4.45}$$

- Do not use *SP trees* not containing arc (h^*, k^*), i.e.,

$$\sum_{T \in \tilde{\Omega} \setminus (\tilde{\Omega}_{t^*} \cap \tilde{\Omega}_{h^* k^*})} \tau_T^{t^*} = 0, \tag{4.46}$$

to create n_r and n_l, respectively. For both branches we just need to modify the upper bounds of the corresponding τ variables. Similarly, for n_r we need to modify the upper bound of the flow variable whereas for n_l we add a new constraint. Alternatively, together with the cutting plane in (4.46), the result of the following proposition can also be used to define n_r.

Proposition 4.5.1. *Suppose that* (h,k) *is an* SP arc *for the pair* (s,t)*. Then, the fraction of* d_{st} *routed on* (h,k) *satisfies the condition*

$$f_{hk}^{st} \geq \frac{1}{deg(s) * \prod_{l \in V \setminus \{s,t\}: deg(l)>1} (deg(l)-1)} \tag{4.47}$$

Proof. Suppose that arc (h,k) is an *SP arc* for the demand pair (s,t). In the worst case the demand d_{st} originated at the source node s would visit all nodes in the graph $G=(V,E)$ before it ceases at the destination node t and all arcs of G would be *SP arcs*. Then, for the source node s we would have

$$f_{sj}^{st} = \frac{1}{deg(s)} \quad \forall (s,j) \in A$$

whereas

$$f_{hk}^{st} = \begin{cases} \frac{\sum_{(l,h)\in A} f_{lh}^{st}}{(deg(h)-1)} & deg(h) > 1 \\ \sum_{(l,h)\in A} f_{lh}^{st} & deg(h) = 1 \end{cases} \quad \forall (h,k) \in A, \;\; h \in V \setminus \{s,t\}$$

for the rest of the graph. For example, suppose that $SP_{st} = \{(s,i),(i,j),(j,v),....,(k,t)\}$ is a shortest path from s to t and all arcs incident to all nodes on this path are *SP arcs*. Then, we have

$$\begin{aligned} f_{si}^{st} &\geq \frac{1}{deg(s)} \\ f_{ij}^{st} &\geq \frac{1}{deg(s) * (deg(i)-1)} \\ f_{jv}^{st} &\geq \frac{1}{deg(s) * (deg(i)-1) * (deg(j)-1)} \\ &\vdots \\ f_{kt}^{st} &\geq \frac{1}{deg(s) * \prod_{l \in PATH_{sk}} (deg(l)-1)} \end{aligned}$$

and hence

$$f_{kt}^{st} \geq \frac{1}{deg(s) * \prod_{l \in V \setminus \{s,t\}: deg(l)>1} (deg(l) - 1)}$$

where $PATH_{sk}$ is the set of nodes on the shortest path from s to k. The latter inequality is based on the assumption that in the worst case d_{st} would visit all nodes before it reaches its destination node. □

In our computational experiments, we have used (4.47) rather than (4.44). This is mainly because our models are already difficult to solve and we do not want the increase the size of current problem as we go down the B&P tree. Moreover, unlike (4.44), the inequalities (4.47) ensure that the flow on an *SP arc* (h, k) is positive. We have observed that this difference has improved the performance of the B&P algorithm for the set of instances we have worked on.

4.6 Computational experiments

In accordance with our previous notation, let $W \subseteq V$ be the set of demand and\or supply nodes, which we call *terminal* nodes and $Q = \{(s,t) \in W \times W : s \neq t\}$ be the set of directed demand pairs with flow demands d_{st}. In order to test our models as well as the B&P algorithm, we have considered the following two well known demand uncertainty definitions we have also mentioned in Chapter 3:

- *Robust box uncertainty à la Bertsimas-Sim (BS)*, which is the box model with a robustness dimension. Suppose that for each pair in Q, the demand d_{st} ranges between d'_{st} and $d'_{st} + \hat{d}_{st}$ (where $\hat{d}_{st} > 0$) and that not more than Γ demands may differ from their nominal values d'_{st} simultaneously. We can define each demand as $d_{st} = d'_{st} + \beta_{st}\hat{d}_{st}$, where β_{st} is a binary variable, and impose that $\sum_{(s,t)\in Q} \beta_{st} \leq \Gamma$. Since $\beta_{st} = \frac{d_{st} - d'_{st}}{\hat{d}_{st}}$, if we relax integrality [3] of

[3]Note that this is legitimate as we have mentioned in Section 3.3.2.

β, the BS uncertainty model defines the polyhedral set of feasible demands as:

$$D = \{d \in \mathbb{R}_+^{|Q|} : d'_{st} \leq d_{st} \leq d'_{st} + \hat{d}_{st} \quad \forall (s,t) \in Q; \ \sum_{(s,t)\in Q} \frac{d_{st} - d'_{st}}{\hat{d}_{st}} \leq \Gamma\}.$$

- *Asymmetric hose model*, with the following definition of feasible demands:

$$D = \{d \in \mathbb{R}_+^{|Q|} : \sum_{t \in W \setminus \{s\}} d_{st} \leq b_s^+; \quad \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s^- \quad \forall s \in W\}$$

where b_s^- and b_s^+ are the ingress and egress capacities of the terminal node $s \in W$, respectively.

4.6.1 Numerical results

We have performed numerical experiments on instances of various sizes to assess the performance of our formulations and the B&P algorithm. We have also included MPLS routing in our estimations to compare it to the OSPF routing with ECMP condition under weight management. Note that the MPLS oblivious performance ratio under general demand uncertainty is found by solving the linear program OB_{POL}, where we restrict neither the routing pattern nor how each demand d_{st} is shared among multiple paths between s and t. Therefore, MPLS routing does not perform worse than OSPF routing with ECMP. Nevertheless, it would be a good benchmark for us to comment on the oblivious ratios under OSPF environment since we have $z_{mpls} \leq z_{ospf}$ where z_{mpls} and z_{ospf} are the oblivious performance ratios for MPLS routing and OSPF routing with ECMP, respectively. Furthermore, Fortz and Thorup [24] compare the performance of optimal OSPF routing with the optimal MPLS routing for a *fixed* TM. They state that although the performance of OSPF routing with weight management is within a few percent of the performance of MPLS routing for a very specific case they consider, it is not possible to generalize this result to any instance. Fortz and Thorup [24] ensure balanced load distribution via a piecewise linear increasing and convex cost function, which depends on link utilizations. However, we deal with oblivious routing where there is a *set* of feasible demands rather than a single TM. To the best of our knowledge, there is no other reference comparing

oblivious MPLS routing with oblivious OSPF routing with weight management for such a general definition of feasible traffic matrices. Therefore, we believe it is important to extend this comparison to the case of a *set* of feasible demands rather than a single TM as Applegate and Cohen [3] also mention.

The instances *bhvac, pacbell, eon, metro*, and *arpanet* are well known instances studied in the IEEE literature. On the other hand, *exodus (Europe), abovenet (US), vnsl (India)*, and *telstra (Australia)* are from the Rocketfuel project [62] for which we have the data for the topology ($|V|$ and $|E|$), the link weights (w), and the number of data packets entering and leaving each node. For these instances, we have assumed that the weight metric w obey the inverse capacity weight setting where the weight of each link is inversely proportional to its capacity, i.e., $c_{hk} = 1/w_{hk} \ \ \forall\{h,k\} \in E$. Moreover, since the information on real demand matrices is not made publicly available, we have used the Gravity model mentioned by Applegate and Cohen [3] to generate the demand polyhedra D matching each instance. This approach is based on the assumption that a demand d_{st} is proportional to the product of a *repulsion* term R_s associated with the source, and an *attraction* term A_t associated with the destination, which for instance can be set as the total observed outgoing and incoming traffic, respectively. A *base demand* $\bar{d}$ is defined and the uncertainty polyhedron is constructed around $\bar{d}$: we have the data on the number of data packets incoming and outgoing for each node h, i.e., the repulsion (R_h) and attraction (A_h) parameters. Then, the base demand for pair (s,t) is estimated using the relation $\bar{d}_{st} = \beta R_s A_t$, where β is computed in order for $\bar{d}$ to be feasible (i.e. to admit at least one routing) and to choose how close $\bar{d}$ is to the boundary of the feasibility region. Let us define $\varsigma \in [0,1]$ such that $\beta = \varsigma \upsilon^*$ with

$$
\begin{aligned}
\upsilon^* = \max \upsilon & \\
\text{s.t.} \quad \sum_{k:\{s,k\}\in E} (g^{st}_{sk} - g^{st}_{ks}) = \upsilon R_s A_t & \quad \forall (s,t) \in Q \\
\sum_{k:\{h,k\}\in E} (g^{st}_{hk} - g^{st}_{kh}) = 0 & \quad \forall h \in V \setminus \{s,t\}, (s,t) \in Q \\
\sum_{(s,t)\in Q} (g^{st}_{hk} + g^{st}_{kh}) \leq C_{hk} & \quad \forall \{h,k\} \in E \\
g^{st}_{hk} \geq 0 & \quad \forall (h,k) \in A, \ (s,t) \in Q.
\end{aligned}
$$

We fix a *direction* (the half-line $\bar{d}_{st} = \beta R_s A_t$) on which $\bar{d}$ must lie, and solve the LP above to find the most critical demand value, which is on the boundary of the feasibility region. Then, ς scales this value so that $\bar{d}$ is an inner point of the demand polyhedron if $\varsigma < 1$. As a result, $[d_{st}]_{(s,t)\in Q}$ is a feasible traffic matrix for the current topology such that the maximum congestion is no more than ς.

For the Hose and BS uncertainty models, we have determined the set of terminal nodes W among the busiest nodes, i.e. the ones with large R_h and A_h parameters. It should be mentioned that our instances are *dense* instances in the sense that in all but two cases we have $|Q|/|V| \geq 0.33$. Moreover, we have created 4 variants of each instance using different uncertainty parameters p with values $\{1.1,\ 2,\ 5,\ 20\}$ for the BS model. We will refer to each BS instance using the label *(name,p)*, i.e., *(nsf,2)* is the nsf instance with uncertainty level $p = 2$. Larger p values imply higher variation in demand estimates. Hence the optimal oblivious ratio is also expected to be larger for such cases. On the other hand, we have randomly picked a subset S of W such that $|S| = \lfloor |W|/2 \rfloor$. Then, we have used $b_s^+ = (\sum_{t\in W\setminus\{s\}} \bar{d}_{st})/1.1\ \forall s \in S$, $b_s^+ = 1.1(\sum_{t\in W\setminus\{s\}} \bar{d}_{st})\ \forall s \in W \setminus S$, $b_s^- = 1.1(\sum_{t\in W\setminus\{s\}} \bar{d}_{st})\ \forall s \in S$, and $b_s^- = (\sum_{t\in W\setminus\{s\}} \bar{d}_{st})/1.1\ \forall s \in W \setminus S$ as the outflow and inflow capacities of the terminal nodes in the hose model. It is worth noting that the uncertainty set is asymmetric in this case. This feature is believed to complicate the problem based on the VPN design literature and our previous experience (Section 2.2 and Chapter 3).

We have used AMPL to model the flow formulation as well as the MPLS routing and Cplex 9.1 MIP solver to solve them. The B&P algorithm is implemented in C using MINTO (Mixed INTeger Optimizer) [59] and Cplex 9.1 as LP solver. We have set a two hours time limit both for AMPL and MINTO. Our test results for two uncertainty models discussed above are summarized in Table 4.1 and Table 4.2 with:

- the instance characteristics, i.e., the name of the instance as well as the numbers of nodes, edges, and terminals;
- the measure p of demand uncertainty that we use in creating the test instances for the BS model. After getting an estimate of the average traffic

demand ($\bar{d}_{st}$) for a pair (s,t), we set the corresponding $d'_{st} = \bar{d}_{st}/p$ and $\hat{d}_{st} = (p - \frac{1}{p})\bar{d}_{st}$;

- the solution z_{tree} and total CPU time t_{tree} of the B&P algorithm;
- the solution z_{flow} and CPU time t_{flow} of the flow formulation;
- the solution z_{mpls} and CPU time t_{mpls} for the MPLS routing.

All run times are given in seconds.

The OSPF routing problem we focus on is clearly different from the regular OSPF routing with fixed link metric. Applegate and Cohen [3] call this more complicated routing effort as *best OSPF style* routing and mention that it is highly non-trivial. Therefore, it is not surprising that some instances could not be solved to optimality at the end of 2 hours time limit. In those cases for which we could find a feasible but not the optimal solution of the corresponding problem, we put a * next to this upper bound. On the other hand, if no feasible solution is available, then the best lower bound obtained by solving the associated LP relaxation is given in brackets. Furthermore, *NoI* means that we do not even have a feasible solution for the LP relaxation, i.e., the Phase I problem could not be solved in 2 hours. Finally, MINTO could not solve some instances due to excessive memory requirements. We label such cases with *mem* under the t_{tree} column.

Note that the z_{tree}, z_{flow}, and z_{mpls} columns provide a relative performance measure for the corresponding routings. They indicate how much each routing deviates from the optimal oblivious routing for the corresponding D in the worst case. Hence, as specified in our mathematical models, these values can be at least 1.0 where larger numbers imply larger deviation from the best possible routing tailored for that instance. Moreover, a value of 1.0 means that the perfectly oblivious routing is found by solving the corresponding model. In other words, by using our optimization tools we find a routing, which is the best tailored for any traffic matrix in the feasible set D. On the other hand, a value larger than 1.0, say 1.2, means that there is at least one feasible TM for which the optimal

OSPF routing performs 20% worse than the best possible routing for that TM but it would not perform any worse than that for any other feasible TM. In other words, the maximum regret for the optimal OSPF routing for D will be 20% when the oblivious ratio is 1.2.

Table 4.1 shows the results for the BS uncertainty model for 11 instances of four different levels of uncertainty. As expected, the oblivious ratios never get smaller as the variability increases. MINTO and Cplex could solve 19 and 17 of these 44 instances to optimality in 2 hours, respectively. Moreover, in the other cases, neither MINTO nor Cplex is clearly superior to the other. For example Cplex outperforms MINTO for the *nsf* instances whereas our B&P method finds the perfectly oblivious OSPF routing for *(example,1.1), (bhvac,1.1), (bhvac,2), (bhvac,5), (abovenet,2), (abovenet,5)*, and *(abovenet,20)* in around one minute. Cplex could only find very loose upper bounds for the *abovenet* instances and just lower bounds for the remaining four. Hence, we can say that it is worth implementing a specialized B&P algorithm for the BS uncertainty model. However, this problem has considerable memory requirements. Therefore, it is not likely to get very promising results for large instances within reasonable time limits neither with the tree nor with the flow formulation. As a result, the performance of optimal oblivious OSPF routing with weight management is not expected to be comparable with the performance of optimal oblivious MPLS routing for large cases under limited computation times.

A comparison of the OSPF and MPLS routings based on our test results should be made in two stages. In the first step, we focus on the 24 instances for which we could find the optimal solutions using either the flow or the tree formulations and compare the gap for the oblivious ratios. In 15 out of 24, we could find the perfectly oblivious routings with both routing protocols. For the remaining 9, the oblivious ratio of our OSPF routing is 5.4% to 47% larger than that of the oblivious MPLS routing. An important observation here is that the gap between two alternatives does not improve with p. In other words, for any network, the deviation for OSPF at uncertainty level p is almost never less than the one for a smaller p. For example, consider the nsf instance for which the oblivious MPLS routing performs strictly better in all of the four uncertainty

Table 4.1: Results for the BS uncertainty model.

name	(V,E,W)	p	z_{tree}	t_{tree}	z_{flow}	t_{flow}	z_{mpls}	t_{mpls}
exodus	(7,6,7)	1.1	1	0.06	1	0.05	1	0.05
		2	1	0.05	1	0.04	1	0.05
		5	1	0.05	1	0.04	1	0.04
		20	1	0.04	1	0.04	1	0.04
nsf	(8,10,5)	1.1	1.17*	2 hrs	1.05*	2 hrs	1.01	0.37
		2	2.05*	2 hrs	1.56	3821.5	1.44	0.75
		5	3.81*	2 hrs	1.90	94.33	1.42	0.98
		20	3.94*	2 hrs	1.98	241.1	1.46	1.05
vnsl	(9,11,3)	1.1	1.07	4.02	1.07	0.19	1	0.02
		2	1.07	23.56	1.07	0.14	1	0.02
		5	1.07	14.67	1.07	0.22	1	0.02
		20	1.07	9.24	1.07	0.30	1	0.02
example	(10,15,4)	1.1	1	0.11	(1)	2 hrs	1	0.28
		2	1	0.15	1	1900.2	1	0.41
		5	2.25*	2 hrs	1.82*	2 hrs	1.03	0.55
		20	2.58*	2 hrs	3.27*	2 hrs	1.08	0.78
metro	(11,42,5)	1.1	4.36*	2 hrs	(1)	2 hrs	1	93
		2	(1.21)	2 hrs	(1.21)	2 hrs	1.21	451
		5	(2.19)	2 hrs	(1.30)	2 hrs	1.30	4642.4
		20	(1.65)	2 hrs	(1.31)	2 hrs	1.30	3577.8
bhvac	(19,22,11)	1.1	1	109.63	(1)	2 hrs	1	81.18
		2	1	120.03	(1)	2 hrs	1	23
		5	1	41.32	(1)	2 hrs	1	44.23
		20	(1.71)	2 hrs	(1)	2 hrs	1.44	1130.5
abovenet	(19,34,5)	1.1	1	12.78	1	60.78	1	12.48
		2	1	13.58	2.24*	2 hrs	1	35.95
		5	1	13.92	2.69*	2 hrs	1	54.06
		20	1	16.31	5.36*	2 hrs	1	46.35
telstra	(44,44,7)	1.1	1	1.75	1	0.50	1	0.16
		2	1	1.79	1	0.41	1	0.16
		5	2.08*	2 hrs	1.05	2.56	1	0.16
		20	2.08*	2 hrs	1.89	2.39	1.28	0.18
pacbell	(15,21,7)	1.1	1.67*	2 hrs	1.28*	2 hrs	1.01	70.93
		2	1.89*	2 hrs	(1.25)	2 hrs	1.25	134
		5	(1.52)	2 hrs	(1.49)	4403	1.49	174.3
		20	(1.57)	2 hrs	(1.54)	2 hrs	1.54	159.5
eon	(19,37,15)	1.1	(1)	2 hrs	NoI	2 hrs	NoI	2 hrs
		2	(1)	2 hrs	NoI	2 hrs	4.43*	2 hrs
		5	(4.72)	2 hrs	NoI	2 hrs	NoI	2 hrs
		20	(6.41)	2 hrs	NoI	2 hrs	6.87*	2 hrs
arpanet	(24,50,10)	1.1	(1.31)	2 hrs	NoI	2 hrs	1.017	492.9
		2	(1.92)	2 hrs	NoI	2 hrs	4.4*	2 hrs
		5	(4.99)	2 hrs	NoI	2 hrs	NoI	2 hrs
		20	(5.8)	2 hrs	NoI	2 hrs	NoI	2 hrs

levels. A comparison of the three routing technologies, namely our *best OSPF style* routing, MPLS routing, and OSPF under inverse capacity weight setting with ECMP, for the *nsf* network is provided in Figure 4.2.

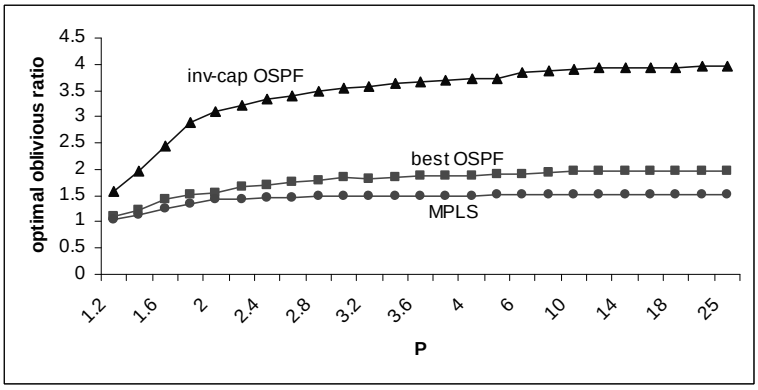

Figure 4.2: The change in the optimal solutions of the *best OSPF style*, MPLS, and inverse capacity weight routings for the network *nsf* for different values of p.

Firstly, notice the significant difference between the *best OSPF style* routing and the OSPF in inverse capacity weight environment. This is a very good example to depict the benefit of using weight management. As it is also clear from Figure 4.2, weight management resulted in an improvement in the OSPF performance. A more concrete comparison of the three alternative routing schemes is given in Figure 4.3, which shows the gaps between the optimal performance ratios.

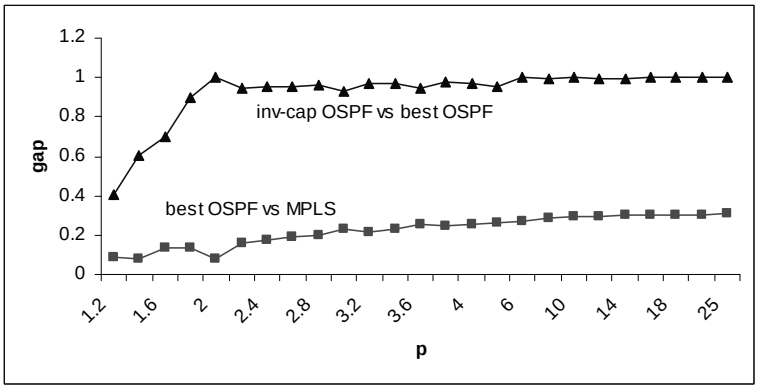

Figure 4.3: Comparison of the *best OSPF style* routing with MPLS and OSPF under inverse capacity weight setting for the instance *nsf* for different values of p.

We can say that inverse capacity OSPF routing is almost 100% worse than *best OSPF* in all higher uncertainty levels for the *nsf* network. On the other hand, the gap between *best OSPF* and MPLS increases with p from 8% to 30%. Finally, due to increasing demand uncertainty, the performances of MPLS, *best OSPF*, and inverse capacity OSPF routings degrade by 32%, 43.6%, and 60.4%, respectively. The degradation in oblivious ratio with uncertainty is already expected. Additionally, these observations certify that the effect is more significant for both OSPF routing strategies. However, we can say that weight management has also helped to reduce the impact of demand uncertainty on oblivious ratio to some extent. Finally, we compare the best upper bounds we obtain for the OSPF routing with the optimal solutions for MPLS. The gaps are more variable for those instances and range from 3.7% to 335.7%. Just like the previous comment, the deviation is larger for more uncertain as well as more difficult[4] instances.

The second traffic uncertainty model we focus on is the hose model for which the test results are shown in Table 4.2. The most obvious comment we can make is that the management of the hose uncertainty model is more difficult than the BS model both for OSPF and MPLS routings. We can make such a comment based on the computation times. Moreover, for the instances *eon* and *arpanet*, we could not get even a feasible solution with neither the flow nor the MPLS formulations. Hence, we believe it will be fair to focus on the other instances of the hose model while interpreting the numerical results.

Performances of the tree and flow formulations in terms of computation times are comparable for relatively smaller instances like *exodus* and *vnsl* where the optimal oblivious ratios are found. Nonetheless, the B&P algorithm had to stop due to excessive memory requirements for *example* and *telstra* providing upper bounds on the optimal oblivious ratios of our *best OSPF style* routing. These bounds are worse than the bounds provided by the flow formulation under the same settings. On the other hand, the tree formulation is superior with respect to the lower bounds found at the end of 2 hours.

[4]We consider large and dense topologies as difficult instances. t_{mpls} values are also indicators of the difficulty level.

Table 4.2: Results for the hose uncertainty model.

name	(V,E,W)	z_{tree}	t_{tree}	z_{flow}	t_{flow}	z_{mpls}	t_{mpls}
exodus	(7,6,7)	1	0.04	1	0.052	1	0.031
nsf	(8,10,5)	4*	2 hrs	2	2730.38	1.52	0.403
vnsl	(9,11,3)	1.07	8.77	1.07	0.296	1	0.16
example	(10,15,4)	2.7*	*mem*	2*	2 hrs	1.08	0.424
metro	(11,42,5)	(1.44)	2 hrs	(1.30)	2 hrs	1.30	1657.83
bhvac	(19,22,11)	(2.85)	2 hrs	(1.52)	2 hrs	(1.52)	2 hrs
abovenet	(19,34,5)	(1.12)	2 hrs	(1.12)	2 hrs	1.05	326.13
telstra	(44,44,7)	2.08*	*mem*	1.93	1.22	1.28	0.08
pacbell	(15,21,7)	(1.55)	2 hrs	(1.54)	2 hrs	1.54	59.13
eon	(19,37,15)	(6.86)	2 hrs	NoI	2 hrs	NoI	2 hrs
arpanet	(24,50,10)	(5.85)	2 hrs	NoI	2 hrs	NoI	2 hrs

The difference between the OSPF and MPLS routings is more evident for the hose model. For *exodus*, we could find the perfectly oblivious routing with both protocols. However, a comparison between the optimal solutions of the instances *nsf, vnsl, example*, and *telstra* shows that the differences between the two alternatives are 31.8%, 6.6%, 85.4%, and 50%, respectively. In brief, the average gap between the optimal solutions of the two routing schemes is 34.8% for the hose model and 6.5% for the BS model. Note that the hose model relies on the estimates for the total inflow and outflow capacities of the routers whereas for the BS case we need an estimate for the lower and upper bounds on the individual demands. Thus, we can say that the definition of the traffic polyhedra D is looser in the former[5] and it is less informative about the possible demand realizations. Therefore, we believe that these average deviations between the two protocols support our remark on higher degradation of network performance due to increased uncertainty for OSPF routing when compared with MPLS.

Our final comment is about the benefit of considering a polyhedra of demands rather than a single traffic matrix $\bar{d}$ of average demands. To make such a comparison we use $\frac{MaxU^{f^*}_{\bar{d}}}{BEST_{\bar{d}}}$, where f^* is the optimal oblivious OSPF routing in a given instance and $BEST_{\bar{d}}$ is the maximum link utilization of the most fair routing,

[5]Based on how we have determined b^+_s and b^-_s as well as d'_{st} and $\hat{d}_{st}$ for the hose and BS instances respectively given the same average pairwise demand estimates $\bar{d}_{st}$.

say $f_{\bar{d}}$, for the average demand $\bar{d}$. First, note that such a comparison does not provide additional information in those instances where we could find the perfectly oblivious routing. We already know that the most fair routing for *any* traffic matrix in D is attained in such cases. Hence we focus on the remaining examples and we have observed that it is not possible to make a conclusion that is valid for all cases. For example in the *vnsl* instances the optimal routing for $\bar{d}$, is different than f^*. This means that if we optimize just for the mean demand and the current demand turns out to be a different one, then we might have $f_{\bar{d}}$ perform significantly worse than f^*. On the other hand, for the nsf instances we have observed that $f_{\bar{d}} \equiv f^*$. As a result, we believe that optimizing just for the mean demands does not suffice to ensure the fair allocation of work load all the time.

4.7 Concluding remarks

Current traffic engineering efforts are mostly based on the efficient use of network resources so as to route a given traffic matrix. However, the demands are not likely to be known exactly in practice. Mulyana and Killat [53] consider inclusion of polyhedral demands with OSPF routing in a rather restricted case since their work applies to a very specific form of uncertainty definition. On the other hand, Applegate and Cohen [3] initiate the works on oblivious routing with demand uncertainty, which has been extended by Belotti and Pınar [11] very recently. Hence, being inspired by the previous works, we consider the case where the polyhedra of feasible demands is defined by some system specific constraints. We incorporate this general uncertainty into the OSPF style routing problem. To comply with the current forwarding technology, we also include the equal load sharing condition (ECMP) in our analysis. Furthermore, we employ weight management to improve the network performance of OSPF. Given all these specifics of the problem we focus on the minimization of the maximum link congestion via a fair allocation of traffic among the network links. To the best of our knowledge this is the first effort to consider these three issues simultaneously and the first work on such a general and practically defensible *best OSPF style* routing.

We have proposed two mixed integer models obtained by a duality-based reformulation for our problem. The first one is a compact formulation based on flow variables. Because this model gets large very rapidly even for medium sized problems, we have proposed an alternative tree formulation based on special structured subgraphs of the backbone graph, i.e., *SP trees*. Moreover, we have proposed a B&P algorithm supported by cutting planes to solve this model.

We have tested our models and the B&P algorithm on two traffic uncertainty definitions, namely the hose model and the BS model. We have presented a comparison of the two formulations in terms of the solution quality and computation times. We have observed that it pays to create a specialized B&P algorithm especially for the BS uncertainty case. Additionally, we have compared the OSPF style routing and the MPLS style routing for these two traffic polyhedra. First, we have realized that for the BS case the optimal oblivious ratios for both routing styles increase as the level of demand variability increases. Another important observation is that the performance of OSPF routing degrades more than the MPLS routing as the demand uncertainty increases. To sum up, we believe that a polyhedral definition of the feasible set of demand matrices, which is accurate as far as possible, could make the OSPF performance get closer to the MPLS performance.

Chapter 5

Robust Network Loading Problem

As the world turns into a 'global village', a tremendous interest in the telecommunications networks is triggered, which has led a notable evolution in the related field. Companies start to embark worldwide collaborations, which imply vast amount of communication traffic at high quality between business partners. Thus, private networks loom large in the contemporary telecommunications networks.

In this chapter we study a fundamental private network design problem, which is encountered in the telecommunications network field: Given a set of possible traffic demands between some endpoints of a customer organization, we need to determine the least costly transmission facility configuration that supports any valid traffic pattern. The most important distinction between the current problem and the VPN design problem in Chapter 3 is that the capacity must be reserved in discrete units and the flow is splittable now.

The rest of this chapter is organized as follows. In Section 5.1, we give a description of our problem along with a compact mixed-integer programming formulation. We develop an elegant decomposition property of our model obtained by projection in Section 5.2. We pass to a specific polyhedral uncertainty model, the hose model [22], in Section 5.3, and carry out a thorough polyhedral

analysis for Network Loading Problem (NLP) under hose uncertainty in Section 5.4. Then we continue with separation algorithms for various valid inequalities and heuristics, all incorporated into a Branch-and-Cut algorithm in Section 5.5. We give an extensive summary of our computational results as well as comparisons with an off-the-shelf mixed-integer solver in Section 5.6 and conclude in Section 5.7 with some directions on future work.

5.1 Problem definition

The deterministic Network Loading Problem (NLP) is defined as follows. Let $G = (V, E)$ be an undirected graph where V is the set of nodes and E is the set of edges. Let Q denote the set of commodities, i.e., $Q \subseteq \{(s,t) : s,t \in V, s \neq t\}$ such that $Q \neq \emptyset$. A set of alternative facility types having different capacities can be used to carry flow through the network. These facilities can only be installed on the edges (but not the nodes) of the network and the problem is about determining the number of facilities to be used on backbone edges such that all demand can be routed at minimum cost. Then, NLP can be modeled as

$$\min \sum_{\{h,k\}\in E} \sum_{l\in L} p^l_{hk} y^l_{hk} \tag{5.1}$$

$$\text{s.t} \sum_{k:\{h,k\}\in E} \left(f^{st}_{hk} - f^{st}_{kh}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \tag{5.2}$$

$$\sum_{(s,t)\in Q} d_{st}(f^{st}_{hk} + f^{st}_{kh}) \leq \sum_{l\in L} C^l y^l_{hk} \qquad \forall \{h,k\} \in E \tag{5.3}$$

$$y^l_{hk} \geq 0 \text{ and integer} \qquad \forall \{h,k\} \in E,\ l \in L \tag{5.4}$$

$$f^{st}_{hk}, f^{st}_{kh} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q \tag{5.5}$$

where d_{st} is the forecasted demand from origin s to destination t for $(s,t) \in Q$, L is the set of facility alternatives, p^l_{hk} is the cost of installing one facility of type $l \in L$ on edge $\{h,k\} \in E$, and C^l is the transmission capacity of type $l \in L$ facility. On the other hand, variables of the model are y^l_{hk} to show the number of type $l \in L$ facilities loaded on the edge $\{h,k\} \in E$ for the flow in both directions

and f_{hk}^{st} to model the fraction of demand for commodity $(s,t) \in Q$ routed on the edge $\{h,k\} \in E$ in the direction from h to k. Constraints (5.2) are the usual flow conservation constraints for each demand pair at each node. Finally the constraints (5.3) are the edge capacity constraints, which ensure that the total capacity installed on each edge is enough to support the total flow on it in both directions.

In accordance with the main vision of this thesis, we are not convinced that deterministic approaches yield designs eligible for realities of the current business environment. Hence, following a similar method to Chapter 4, rather than building our solution for a specific demand forecast, we use a general demand uncertainty definition with a motivation to design a network that is viable for any demand realization in the polyhedral set

$$D = \{d \in \mathbb{R}^{|Q|} : Ad \leq a,\ d \geq 0\} \tag{5.6}$$

where $A \in \mathbb{R}^{m \times |Q|}$ and $a \in \mathbb{R}^m$ with m being the number of linear inequalities that define the bounded and nonempty polyhedron D. This leads to the following polyhedral NLP model (NLP_{POL}):

$$\min \sum_{\{h,k\} \in E} \sum_{l \in L} p_{hk}^l y_{hk}^l$$

$$\text{s.t} \sum_{k:\{h,k\} \in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q$$

$$\max_{d \in D} \sum_{(s,t) \in Q} d_{st}(f_{hk}^{st} + f_{kh}^{st}) \leq \sum_{l \in L} C^l y_{hk}^l \qquad \forall \{h,k\} \in E \tag{5.7}$$

$$y_{hk}^l \geq 0 \text{ and integer} \qquad \forall \{h,k\} \in E,\ l \in L \tag{5.8}$$

$$f_{hk}^{st}, f_{kh}^{st} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q \tag{5.9}$$

where f_{hk}^{st} is the fraction of the demand of commodity (s,t) routed on the directed arc (h,k).

Unlike the case with known demands, NLP_{POL} is a semi-infinite optimization model due to the maximization problem we need to solve over the demand polyhedron for each edge $\{h,k\} \in E$ in (5.7). Next, following the method we have

used both in Proposition 3.3.1 and Proposition 4.3.1, we give a compact linear MIP formulation for NLP_{POL}.

Proposition 5.1.1. *Assuming that demand is subject to polyhedral uncertainty as in (5.6), NLP_{POL} can be reformulated as the following linear MIP model (NLP_{GD}):*

$$\min \sum_{\{h,k\}\in E} \sum_{l\in L} p^l_{hk} y^l_{hk} \tag{5.10}$$

$$s.t \quad (5.2),\ (5.4),\ (5.5)$$

$$\sum_{z=1}^{m} a_z \lambda^{hk}_z \leq \sum_{l\in L} C^l y^l_{hk} \quad \forall \{h,k\} \in E \tag{5.11}$$

$$f^{st}_{hk} + f^{st}_{kh} \leq \sum_{z=1}^{m} a^{st}_z \lambda^{hk}_z \quad \forall (s,t) \in Q,\ \{h,k\} \in E \tag{5.12}$$

$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1,...,m,\ \{h,k\} \in E. \tag{5.13}$$

Proof. Consider NLP_{POL}. Then for a given flow vector f and edge $\{h,k\} \in E$ the worst-case capacity requirement can be found by solving the following problem $P(h,k)$:

$$\begin{aligned} &\max \sum_{(s,t)\in Q} (f^{st}_{hk} + f^{st}_{kh}) d_{st} \\ \text{s.t} \quad &\sum_{(s,t)\in Q} a^{st}_z d_{st} \leq a_z \qquad \forall z = 1,...,m \\ &d_{st} \geq 0 \qquad \forall (s,t) \in Q. \end{aligned} \tag{5.14}$$

Notice that $P(h,k)$ is a linear programming model and its dual is $D(h,k)$:

$$\begin{aligned} &\min \sum_{z=1}^{m} a_z \lambda^{hk}_z \\ \text{s.t} \quad &\sum_{z=1}^{m} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \quad \forall (s,t) \in Q \\ &\lambda^{hk}_z \geq 0 \qquad \forall z = 1,...,m, \end{aligned}$$

where λ^{hk}_z is the dual variable corresponding to (5.14). Since $P(h,k)$ is feasible and bounded, we can use duality transformation similar to the one of Soyster [61].

Hence for each edge $\{h,k\} \in E$, we can replace (5.7) with

$$\sum_{l \in L} C^l y^l_{hk} \geq \min \sum_{z=1}^{m} a_z \lambda^{hk}_z$$
$$\text{s.t} \quad \sum_{z=1}^{m} a^{st}_z \lambda^{hk}_z \geq f^{st}_{hk} + f^{st}_{kh} \quad \forall (s,t) \in Q$$
$$\lambda^{hk}_z \geq 0 \qquad \forall z = 1, ..., m.$$

Then, we can omit the *min* since the objective function (5.10) tries to minimize the sum of the design variables y^l_{hk} with nonnegative weights. □

5.2 Projecting out the flow variables

In this section, we study the projection of NLP_{GD} on to the space of $\lambda \in \mathbb{R}^{|W||E|}$ and design variables $y \in \mathbb{Z}^{|L||E|}$, which is possible since there is no flow cost in our model. Firstly, we replace the flow conservation constraints (5.2) with equivalent inequality forms

$$\sum_{k:\{h,k\} \in E} \left(f^{st}_{hk} - f^{st}_{kh} \right) \geq \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q. \quad (5.15)$$

following Mirchandani [52]. Then, we can state the following result.

Proposition 5.2.1. *For a given pair $(\bar{\lambda}, \bar{y})$, there exists a feasible flow $f \geq 0$ for NLP_{GD} satisfying (5.5), (5.12), and (5.15) if and only if*

$$\sum_{(s,t) \in Q} [-\alpha^{st}_s + \alpha^{st}_t + \sum_{\{h,k\} \in E} \beta^{st}_{hk} \sum_{z=1}^{m} (a^{st}_z \bar{\lambda}^{hk}_z)] \geq 0 \quad (5.16)$$

for all $(\alpha, \beta) \geq 0$ such that

$$\alpha^{st}_h - \alpha^{st}_k + \beta^{st}_{hk} \geq 0 \quad \forall (s,t) \in Q, \ \{h,k\} \in E \quad (5.17)$$
$$-\alpha^{st}_h + \alpha^{st}_k + \beta^{st}_{hk} \geq 0 \quad \forall (s,t) \in Q, \ \{h,k\} \in E. \quad (5.18)$$

Proof. For a given $(\bar{\lambda}, \bar{y})$, there exists a feasible flow of all commodities simultaneously if and only if $\{f \in \mathbb{R}^{2|E||Q|} : (5.5), (5.12), (5.15)\} \neq \emptyset$. So, Farkas' Lemma states that the system

$$\textbf{(I)} \quad \sum_{k:\{h,k\}\in E} \left(f_{hk}^{st} - f_{kh}^{st}\right) - e_h^{st} = \begin{cases} 1 & h = s \\ -1 & h = t \\ 0 & \text{otherwise} \end{cases} \qquad \forall h \in V, (s,t) \in Q \quad (5.19)$$

$$-f_{hk}^{st} - f_{kh}^{st} - e_{hk}^{st} = -\sum_{z=1}^{m} a_z^{st} \bar{\lambda}_z^{hk} \qquad \forall \{h,k\} \in E, (s,t) \in Q \quad (5.20)$$

$$f_{hk}^{st}, f_{kh}^{st} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

is feasible if and only if the following system

$$\textbf{(II)} \quad \sum_{(s,t)\in Q} [\alpha_s^{st} - \alpha_t^{st} - \sum_{\{h,k\}\in E} \beta_{hk}^{st} \sum_{z=1}^{m} (a_z^{st} \bar{\lambda}_z^{hk})] > 0$$

$$\alpha_h^{st} - \alpha_k^{st} - \beta_{hk}^{st} \leq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

$$-\alpha_h^{st} + \alpha_k^{st} - \beta_{hk}^{st} \leq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

$$-\alpha_h^{st} \leq 0 \qquad \forall h \in V,\ (s,t) \in Q$$

$$-\beta_{hk}^{st} \leq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

is infeasible where we associate the dual variables α_h^{st} and β_{hk}^{st} with (5.19) and (5.20). The feasibility of (II) can be determined by solving the LP

$$\textbf{(III)} \quad \min \sum_{(s,t)\in Q} [-\alpha_s^{st} + \alpha_t^{st} + \sum_{\{h,k\}\in E} \beta_{hk}^{st} \sum_{z=1}^{H} (a_z^{st} \bar{\lambda}_z^{hk})]$$

$$\text{st.} \quad -\alpha_h^{st} + \alpha_k^{st} + \beta_{hk}^{st} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

$$\alpha_h^{st} - \alpha_k^{st} + \beta_{hk}^{st} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q$$

$$\alpha_h^{st} \geq 0 \qquad \forall h \in V,\ (s,t) \in Q$$

$$\beta_{hk}^{st} \geq 0 \qquad \forall \{h,k\} \in E,\ (s,t) \in Q.$$

If the optimal solution value of (III) is nonnegative, then (II) is infeasible and there exists a feasible multi-commodity flow as we wanted to show. □

A similar projection is obtained in Mirchandani [52] both for single- and multi-commodity NLP with deterministic demands. Although he could characterize all

extreme rays of the related projection cone for the single-commodity case, he could only give necessary conditions for the multi-commodity variant. The difficulty of the latter problem is due to the bundle constraints (5.3), which prevent the decomposition of the problem into single commodity subproblems. However, the situation is completely different for NLP_{GD} as Proposition 5.2.1 reveals an unexpectedly nice property of our problem. The result states that the existence of a multi-commodity flow f can be certified by checking the existence of $|Q|$ single commodity flows, i.e., the projection cone in (5.17)-(5.18) can be decomposed into $|Q|$ smaller cones with one cone for each commodity $(s,t) \in Q$. Thus, we will have the set

$$\begin{aligned} S_{st} = \{\alpha^{st} \in \Re^{|V|}, \beta^{st} \in \Re^{|E|} : \beta^{st}_{hk} \geq -\alpha^{st}_h + \alpha^{st}_k \quad & \forall \{h,k\} \in E \\ \beta^{st}_{hk} \geq \alpha^{st}_h - \alpha^{st}_k \quad & \forall \{h,k\} \in E \\ \alpha^{st}_h \geq 0 \quad & \forall h \in V \\ \beta^{st}_{hk} \geq 0 \quad & \forall \{h,k\} \in E\} \end{aligned}$$

for each commodity $(s,t) \in Q$ and Mirchandani [52] has fully characterized the extreme rays of S_{st}.

For $S \subset V$, let $\delta(S)$ denote the set of edges with exactly one endpoint in S. Moreover, for ease of notation, we will denote an edge $\{h,k\}$ as e when there is no need to know its endpoints.

Proposition 5.2.2. ***[Mirchandani [52]]****Let $(\alpha^{st}, \beta^{st})$ be a ray of the pointed finitely generated cone S_{st}. Then, $(\alpha^{st}, \beta^{st})$ is an extreme ray of S_{st} if and only if it is in one of the following forms:*

- Node Form: $\alpha^{st}_h = 1$ *for all* $h \in V$ *and* $\beta^{st}_e = 0$ *for all* $e \in E$.
- Arc Form: $\alpha^{st}_h = 0$ *for all* $h \in V$ *and* $\beta^{st}_{\bar{e}} = 1$ *for some* $\bar{e} \in E$ *and all other entries are 0.*
- Cutset Form: $\alpha^{st}_h = 1$ *for all* $h \in S$ *and* $\alpha^{st}_h = 0$ *for all* $h \in V \backslash S$. $\beta^{st}_e = 1$ *if* $e \in \delta(S)$ *and 0 otherwise, where* $S \subset V$ *and the subgraph induced by* S *is connected.*

Then we state the valid inequalities implied by each form of extreme rays in the following remark using Proposition 5.2.2.

Remark 2. *The inequalities of type (5.16) implied by each form of the extreme rays are as follows:*

- Node Form: $0 \geq 0$
- Arc Form: $\sum_{z=1}^{m} a_z^{st} \lambda_z^e \geq 0 \;\; \forall e \in E, \; \forall (s,t) \in Q$
- Cutset Form:

$$\sum_{e \in \delta(S)} \sum_{z=1}^{m} a_z^{st} \lambda_z^e \geq \begin{cases} 1 & \text{if } s \in S, \; t \in T \\ -1 & \text{if } s \in T, \; t \in S \\ 0 & \text{otherwise} \end{cases} \quad \forall (s,t) \in Q, \; S \subset V, \; T = V \backslash S.$$

The Node Form implies a redundant inequality. Similarly the inequalities obtained by using the Arc Form or the Cutset Form may become redundant depending on the entries a_z^{st} in (5.6). For example, if all a_z^{st} are nonnegative, then the Arc Form valid inequalities as well as the last two inequalities of the Cutset Form mentioned above are redundant due to the nonnegativity of the λ variables.

Notice that the inequalities listed in Remark 2 are actually the feasibility cuts and must be included in the projection of NLP_{GD} onto the space of (λ, y) variables. Then, the corresponding mathematical model (NLP_{PRO}) in the space of λ and y variables is

$$\min \sum_{e \in E} \sum_{l \in L} p_e^l y_e^l$$

$$\text{s.t} \quad \sum_{z=1}^{m} a_z^{st} \lambda_z^e \geq 0 \qquad \forall e \in E, \; (s,t) \in Q$$

$$\sum_{e \in \delta(S)} \Big(\sum_{z=1}^{m} a_z^{st} \lambda_z^e\Big) \geq \begin{cases} 1 & \text{if } s \in S, \; t \in T \\ -1 & \text{if } s \in T, \; t \in S \\ 0 & \text{otherwise} \end{cases} \quad \forall (s,t) \in Q, \; S \subset V, \; T = V \backslash S$$

$$(5.4), \; (5.11), \; (5.13).$$

Clearly, NLP_{PRO} is a very general model since it can be used for any polyhedral demand definition with $|L|$ facility alternatives. In the sequel, we will first consider the well-known hose model of Duffield et al. [22] as the demand polyhedron, and present results on the polyhedral analysis of the problem for this uncertainty definition and multiple facility types.

5.3 NLP under hose demand uncertainty

From now on, we will consider the symmetric hose model with the polyhedra of demands U_{SymG} defined as

$$D = \{d \in \mathbb{R}^{|Q|} : \sum_{t \in W \setminus \{s\}} d_{st} + \sum_{t \in W \setminus \{s\}} d_{ts} \leq b_s \; \forall s \in W, \;\; d_{st} \geq 0 \quad \forall (s,t) \in Q\} \tag{5.21}$$

where $W \subseteq V$ is the set of nodes in need of communication, $Q = \{(s,t) \in W \times W : s \neq t\}$ is the set of commodities, and b_s is the bandwidth capacity of the terminal node $s \in W$. The importance of the hose model can be demonstrated by returning to the simple example of Chapter 2 on Figure 2.1, where we consider a single facility type with unit capacity and the number on each edge is the capacity installation cost for that edge. Recall that the optimal capacity allocation would be as shown in Figure 2.1b with a total cost of 4 when the demands are assumed to be known. On the other hand, in the corresponding hose model each node would have a bandwidth of 4 units since this is the total flow incident to each node for the given demand forecast. Then, the optimal design for the hose polyhedron is as shown in Figure 5.1 again with the same total cost. The important point here is that although the total design costs are the same for the two cases, the polyhedral design is more robust to fluctuations in demand. Under the scenario, as considered in Chapter 2, that the demands from node A to nodes B and C are realized to be 0.999 and 1.001, the deterministic design becomes unusable although the robust one remains operational. This example illustrates a very important advantage of using the hose uncertainty model. As we have mentioned before using the hose specification, the network service provider

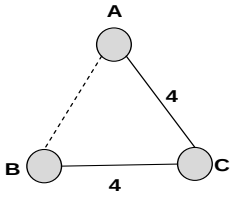

Figure 5.1: Minimum cost design for the hose model.

would have a design flexible to accommodate any arbitrary distribution of the total traffic provided that the total bandwidth capacity is not violated. In other words, it enables the transfer of unused capacity for a pairwise demand to another demand which goes beyond its estimation. Hence, the hose model yields a more flexible design than the deterministic one even for the very simple example above. To the best of our knowledge, this is the first effort to incorporate polyhedral uncertainty other than Karaşan et al. [44], where the emphasis is more on modeling rather than polyhedral properties. Hence, we believe that using U_{SymG} is already a good starting point to initiate future research interests on NLP with different polyhedral uncertainty definitions.

Proposition 5.3.1. *The projection of NLP_{GD} on to the space of (λ, y) variables for the hose model (NLP_{hose}) is as follows:*

$$\begin{aligned} & \min \sum_{e\in E}\sum_{l\in L} p_e^l y_e^l \\ s.t \quad & \sum_{s\in W} b_s \lambda_s^e \leq \sum_{l\in L} C^l y_e^l \quad \forall e \in E & (5.22) \\ & \sum_{e\in\delta(S)} (\lambda_s^e + \lambda_t^e) \geq 1 \quad \forall (s,t) \in Q, S \subset V : s \in S, t \in V\backslash S & (5.23) \\ & (5.4), (5.13) \end{aligned}$$

Proof. The valid inequalities implied by the Arc Form extreme rays, reduce to

$$\lambda_s^e + \lambda_t^e \geq 0 \ \ \forall e \in E, \ (s,t) \in Q. \tag{5.24}$$

Moreover, the Cut Form valid inequalities are

$$\sum_{e\in\delta(S)} (\lambda_s^e + \lambda_t^e) \geq \begin{cases} 1 & \text{if } s \in S,\ t \in T \\ -1 & \text{if } s \in T,\ t \in S \\ 0 & \text{otherwise} \end{cases} \quad \forall S \subset V,\ T = V \backslash S. \tag{5.25}$$

However, the nonnegativity constraints (5.13) dominate (5.24) and the last two inequalities in (5.25). Consequently, NLP_{PRO} reduces to NLP_{hose}, which completes the proof. □

5.4 Polyhedral analysis

In this section we present results on the facets of the polyhedron associated with the network loading problem under hose uncertainty NLP_{hose}.

In the sequel, we assume that

i. C^l is a positive integer for $l \in L$,

ii. for l_1 and l_2 in L such that $l_1 < l_2$ we have $C^{l_1} < C^{l_2}$.

Let $F = \{(\lambda, y) \in \mathbb{R}_+^{|W||E|} \times \mathbb{Z}_+^{|E||L|} : \text{ (5.22) and (5.23)}\}$ and $P = conv(F)$. Observe that adding constraints

$$\lambda_s^e \leq 1 \qquad \forall s \in W, e \in E \tag{5.26}$$

does not change the validity of the model as mentioned in Karaşan et al. [44]. Let $F' = F \cap \{(\lambda, y) \in \mathbb{R}_+^{|W||E|} \times \mathbb{Z}_+^{|E||L|} : \text{ (5.26)}\}$ and $P' = conv(F')$.

For $s \in W$ and $e \in E$, let u_s^e be a unit vector of dimension $|W||E|$ where the entry corresponding to e and s is 1 and other entries are zero. For $e \in E$ and $l \in L$, let v_e^l be a unit vector of dimension $|E||L|$ where the entry corresponding to e and l is one and other entries are zero. Let M be a very large integer.

First, we investigate the dimension of the polyhedra P and P'.

Proposition 5.4.1. *Dimension of P and $P^{'}$ is $(|W| + |L|)|E|$.*

Proof. Suppose that all points in P (resp. $P^{'}$) satisfy $\alpha\lambda + \beta y = \gamma$. Let $e \in E$, $l \in L$, and $(\lambda, y) \in P$ (resp. $(\lambda, y) \in P^{'}$). Observe that the vector $(\lambda, y + v_e^l)$ is also in P (resp. in $P^{'}$). Hence $\beta_e^l = 0$. Let $s \in W$ and $e \in E$. Consider the vectors (λ, y) where $\lambda = \sum_{s' \in W} \sum_{e' \in E} u_{s'}^{e'} - u_s^e$ and $y = \sum_{l \in L} \sum_{e' \in E} M v_{e'}^l$ and $(\lambda + u_s^e, y)$. Since these two vectors are in P (resp. in $P^{'}$), we have $\alpha_s^e = 0$. As there is no equality that is satisfied by all points in P (resp. in $P^{'}$), polyhedron P (resp. $P^{'}$) is full-dimensional. □

Next, we will state the relationship between the facet defining inequalities of P, P', and their projection on to the space of λ variables. But for the sake of completeness, we should first give the following definition for the projection of a polyhedron to a lower dimensional space.

Definition 5.4.1. *Given a polyhedron $P = \{x \in \mathbb{R}^n,\ y \in \mathbb{R}^m : Ax + By \leq b\}$ where $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$, and $b \in \mathbb{R}^p$, the projection of P onto the space of y variables is*

$$Proj_y(P) = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \text{ such that } (x, y) \in P\}.$$

Consequently, let $F_\lambda = Proj_\lambda(F) = \{\lambda \in \mathbb{R}_+^{|W||E|} : (5.23)\}$ and $F_\lambda^{'} = Proj_\lambda(F^{'}) = F_\lambda \cap \{\lambda \in \mathbb{R}_+^{|W||E|} : (5.26)\}$. Notice that since P (resp. P') is full-dimensional, so is F_λ (resp. $F_\lambda^{'}$) ([8]), i.e., $dim(F_\lambda) = dim(F_\lambda^{'}) = |W||E|$. Now, we relate facet defining inequalities of F_λ and $F_\lambda^{'}$ with those of P and $P^{'}$.

Proposition 5.4.2. *Inequality $\sigma\lambda \geq \sigma_0$ is facet defining for P (resp. for $P^{'}$) if and only if it is facet defining for F_λ (resp. for $F_\lambda^{'}$).*

Proof. If inequality $\sigma\lambda \geq \sigma_0$ is facet defining for P (resp. for $P^{'}$) then it is facet defining for F_λ (resp. for $F_\lambda^{'}$) since $F_\lambda = Proj_\lambda(F)$ and both P and F_λ are full-dimensional (resp. $F_\lambda^{'} = Proj_\lambda(F^{'})$ and both $P^{'}$ and $F_\lambda^{'}$ are full-dimensional). Suppose that inequality $\sigma\lambda \geq \sigma_0$ is facet defining for F_λ (resp. for $F_\lambda^{'}$) and all points in P (resp. in $P^{'}$) such that $\sigma\lambda = \sigma_0$ also satisfy $\alpha\lambda + \beta y = \gamma$. Let $e \in E$,

$l \in L$, and $(\lambda, y) \in P$ (resp. in P') such that $\sigma\lambda = \sigma_0$. The vector $(\lambda, y + v_e^l)$ is also in P (resp. in P') and satisfies $\sigma\lambda = \sigma_0$. Hence $\beta_e^l = 0$. As $\sigma\lambda \geq \sigma_0$ is facet defining for F_λ (resp. for F_λ') and $F_\lambda = Proj_\lambda(F)$ (resp. $F_\lambda' = Proj_\lambda(F')$), (α, γ) is a multiple of (σ, σ_0). Thus inequality $\sigma\lambda \geq \sigma_0$ is facet defining for P (resp. for P'). □

Results similar to Propositions 5.4.1 and 5.4.2, based on projection of polyhedra, are given in Labbé and Yaman [46].

For $e \in E$, define $F_e = \{(\lambda^e, y_e) \in \mathbb{R}_+^{|W|} \times \mathbb{Z}_+^{|L|} : \ (5.22)\}$, $P_e = conv(F_e)$, $F_e' = F_e \cap \{(\lambda^e, y_e) \in \mathbb{R}_+^{|W|} \times \mathbb{Z}_+^{|L|} : \ (5.26)\}$, and $P_e' = conv(F_e')$.

Remark 3. *Observe that if $\delta(S) \setminus \{e\} \neq \emptyset$ for every $S \subset V$ such that there exists $(s,t) \in Q$ with $s \in S$ and $t \in V \setminus S$, then $F_e = Proj_{(\lambda^e, y_e)}(F)$ and $F_e' = Proj_{(\lambda^e, y_e)}(F')$. This follows directly from Definition 5.4.1. For a given pair $(\lambda^e, y_e) \in F_e$, we can always find a pair $(\hat{\lambda}, \hat{y}) \in F$ such that $\hat{\lambda}_s^e = \lambda_s^e$ for all $s \in W$ with $\sum_{e' \in \delta(S)\setminus\{e\}} (\hat{\lambda}_s^{e'} + \hat{\lambda}_t^{e'}) \geq 1$ for all $(s,t) \in Q$, $S \subset V$ such that $s \in S$, $t \in V \setminus S$, and $\hat{y} = \sum_{l \in L} y_e v_e^l + \sum_{l \in L} \sum_{e' \in E \setminus \{e\}} M v_{e'}^l$.*

In the following proposition, we investigate how the facet defining inequalities of P_e and P_e' are related to those of P and P'.

Theorem 1. *Let $e \in E$ be such that $\delta(S) \setminus \{e\} \neq \emptyset$ for every $S \subset V$ such that there exists $(s,t) \in Q$ with $s \in S$ and $t \in V \setminus S$. Inequality $\alpha\lambda^e + \beta y_e \geq \gamma$ is facet defining for P_e (resp. for P_e') if and only if it is facet defining for P (resp. for P').*

Proof. Let $e \in E$ be such that $\delta(S) \setminus \{e\} \neq \emptyset$ for every $S \subset V$ such that there exists $(s,t) \in Q$ with $s \in S$ and $t \in V \setminus S$. If inequality $\alpha\lambda^e + \beta y_e \geq \gamma$ is facet defining for P (resp. for P') then it defines a facet of P_e (resp. for P_e') as $F_e = Proj_{(\lambda^e, y_e)}(F)$ and both P_e and P are full-dimensional (resp. $F_e' = Proj_{(\lambda^e, y_e)}(F')$ and P_e' and P' are full-dimensional).

Suppose that inequality $\alpha\lambda^e + \beta y_e \geq \gamma$ is facet defining for P_e (resp. for P_e'). Suppose also that all points in P (resp. in P') such that $\alpha\lambda^e + \beta y_e = \gamma$ also

satisfy $\overline{\alpha}\lambda + \overline{\beta}y = \overline{\gamma}$. Let $e^{'} \in E \setminus \{e\}$, $l \in L$, and $(\lambda, y) \in P$ (resp. in $P^{'}$) be such that $\alpha\lambda^e + \beta y_e = \gamma$. The vector $(\lambda, \overline{y})$ where $\overline{y} = y + v^l_{e^{'}}$ is also in P (resp. in $P^{'}$) and satisfies $\alpha\lambda^e + \beta\overline{y}_e = \gamma$. Hence $\overline{\beta}^l_{e^{'}} = 0$.

Let $s^{'} \in W$ and $e^{'} \in E \setminus \{e\}$. Let (λ^e, y_e) be a vector in P_e (resp. in $P^{'}_e$) such that $\alpha\lambda^e + \beta y_e = \gamma$. Consider $(\overline{\lambda}, \overline{y})$ where $\overline{\lambda} = \sum_{s\in W} \lambda^e_s u^e_s + \sum_{s\in W} \sum_{e^{''}\in E\setminus\{e\}} u^{e^{''}}_s - u^{e^{'}}_{s^{'}}$ and $\overline{y} = \sum_{l\in L} \sum_{e^{''}\in E\setminus\{e\}} M v^l_{e^{''}} + \sum_{l\in L} y^l_e v^l_e$. Since $\delta(S) \setminus \{e\} \neq \emptyset$ for every $S \subset V$ such that $(s,t) \in Q$ with $s \in S$ and $t \in V \setminus S$, this vector is in P (resp. in $P^{'}$) and satisfies $\alpha\overline{\lambda}^e + \beta\overline{y}_e = \gamma$. Consider also $(\hat{\lambda}, \hat{y})$ where $\hat{\lambda} = \overline{\lambda} + u^{e^{'}}_{s^{'}}$ and $\hat{y} = \overline{y}$. This vector is also in P (resp. in $P^{'}$) and satisfies $\alpha\hat{\lambda}^e + \beta\hat{y}_e = \gamma$. Hence $\overline{\alpha}^{e^{'}}_{s^{'}} = 0$.

Now we can conclude that $\overline{\alpha}\lambda + \overline{\beta}y = \overline{\gamma}$ is a multiple of $\alpha\lambda^e + \beta y_e = \gamma$ as $F_e = Proj_{(\lambda^e, y_e)}(F)$ (resp. $F^{'}_e = Proj_{(\lambda^e, y_e)}(F^{'})$), which completes the proof. □

For $S \subseteq V$, define $b(S) = \sum_{s\in S\cap W} b_s$ and $B(S) = \min\{b(S), b(V \setminus S)\}$. Notice that in the worst case all terminals in $S \subset V$ would want to use all of their bandwidths to exchange traffic with the nodes in $V \setminus S$. As a result, the worst case traffic on the cut $\delta(S)$ would be the minimum of these requirements, i.e., $B(S)$.

Let $S \subset V$ be such that the subgraphs induced by S and $V \setminus S$ are both connected. Let $y_{\delta(S)}$ be the restriction of the vector y to edges $e \in \delta(S)$, $F(S) = \{y_{\delta(S)} \in \mathbb{Z}_+^{|\delta(S)||L|} : \sum_{l\in L} \sum_{e\in\delta(S)} C^l y^l_e \geq B(S)\}$ and $P(S) = conv(F(S))$.

Proposition 5.4.3. *Let $S \subset V$ be such that the subgraphs induced by S and $V \setminus S$ are both connected and $B(S) > 0$. Then, $F(S) = Proj_{y_{\delta(S)}}(F) = Proj_{y_{\delta(S)}}(F^{'})$.*

Proof. Suppose that $S \subset V$ such that the subgraphs induced by S and $V \setminus S$ are both connected and $B(S) > 0$. Let $y_{\delta(S)} \in F(S)$ and assume that $B(S) = b(S)$. The proof for the other case, i.e., $B(S) = b(V \setminus S)$, is similar. Define $(\hat{\lambda}, \hat{y})$ as follows: $\hat{\lambda}^e_s = 1$ for $e \in E \setminus \delta(S)$ and $s \in W$, $\hat{\lambda}^e_s = 0$ for $e \in \delta(S)$ and $s \in W \setminus S$, $\hat{y} = \sum_{l\in L} \sum_{e\in\delta(S)} y^l_e v^l_e + \sum_{l\in L} \sum_{e\in E\setminus\delta(S)} M v^l_e$. Notice that since the subgraphs induced by S and $V \setminus S$ are both connected, for $S^{'} \subset V$ such that $S^{'} \neq S$,

$S' \neq V \setminus S$, and $\delta(S') \setminus \delta(S) \neq \emptyset$. Hence, if $(\hat{\lambda}, \hat{y})$ satisfies

$$\sum_{e \in \delta(S)} \hat{\lambda}_s^e = 1 \qquad \forall s \in S \cap W \tag{5.27}$$

$$\sum_{s \in S \cap W} b_s \hat{\lambda}_s^e \leq \sum_{l \in L} C^l \hat{y}_e^l \qquad \forall e \in \delta(S) \tag{5.28}$$

$$\hat{\lambda}_s^e \geq 0 \qquad \forall s \in S \cap W, e \in \delta(S) \tag{5.29}$$

then $(\hat{\lambda}, \hat{y})$ is in P and P'. By Farkas Lemma, this system is feasible if and only if

$$\sum_{s \in S \cap W} \alpha_s + \sum_{e \in \delta(S)} \sum_{l \in L} C^l \hat{y}_e^l \beta_e \geq 0 \tag{5.30}$$

for all (α, β) such that

$$\alpha_s + b_s \beta_e \geq 0 \qquad \forall s \in S \cap W, e \in \delta(S)$$
$$\beta_e \geq 0 \qquad \forall e \in \delta(S).$$

Since $\sum_{l \in L} C^l \hat{y}_e^l \geq 0$ for all $e \in \delta(S)$, we can limit ourselves to (α, β) such that $\beta_e = \max_{s \in S \cap W} \left(\frac{-\alpha_s}{b_s} \right)^+$ for all $e \in \delta(S)$. Now, we can rewrite the left hand side of (5.30) as

$$\sum_{s \in S \cap W} \alpha_s + \sum_{e \in \delta(S)} \sum_{l \in L} C^l \hat{y}_e^l \max_{s \in S \cap W} \left(\frac{-\alpha_s}{b_s} \right)^+. \tag{5.31}$$

If $\max_{s \in S \cap W} \left(\frac{-\alpha_s}{b_s} \right)^+ = 0$, then $\alpha_s \geq 0$ for all $s \in S \cap W$, and (5.31) is nonnegative. Suppose now that $\max_{s \in S \cap W} \left(\frac{-\alpha_s}{b_s} \right)^+ = \frac{-\alpha_{s'}}{b_{s'}} > 0$ for some $s' \in S \cap W$. Then, (5.31) is

$$\sum_{s \in S \cap W} \alpha_s - \frac{\alpha_{s'}}{b_{s'}} \sum_{e \in \delta(S)} \sum_{l \in L} C^l \hat{y}_e^l. \tag{5.32}$$

Now as $\sum_{e \in \delta(S)} \sum_{l \in L} C^l \hat{y}_e^l \geq b(S)$ and $\frac{-\alpha_{s'}}{b_{s'}} > 0$, (5.32) is greater than or equal to

$$\sum_{s \in S \cap W} \alpha_s - \frac{\alpha_{s'}}{b_{s'}} b(S) = \sum_{s \in S \cap W} (\alpha_s - \frac{\alpha_{s'}}{b_{s'}} b_s).$$

As $\frac{\alpha_s}{b_s} \geq \frac{\alpha_{s'}}{b_{s'}}$ for all $s \in S \cap W$, $\sum_{s \in S \cap W}(\alpha_s - \frac{\alpha_{s'}}{b_{s'}} b_s) \geq 0$. So the system (5.27)-(5.29) always has a solution. Hence, we can always find a $(\hat{\lambda}, \hat{y}) \in F$ such that $(\hat{\lambda}, \hat{y}) \in F'$ for any $y_{\delta(S)} \in F(S)$, which is constructed just as above. This implies that $F(S) \subseteq Proj_{y_{\delta(S)}}(F)$ and $F(S) \subseteq Proj_{y_{\delta(S)}}(F')$. On the other hand, since $\sum_{l \in L} \sum_{e \in \delta(S)} C^l y_e^l \geq B(S)$ is a valid inequality for F and F', $Proj_{y_{\delta(S)}}(F) \subseteq F(S)$ and $Proj_{y_{\delta(S)}}(F') \subseteq F(S)$. □

Now, we can relate facet defining inequalities of $P(S)$ to those of P.

Theorem 2. *Let $S \subset V$ be such that the subgraphs induced by S and $V \setminus S$ are both connected and $B(S) > 0$. If inequality $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l \geq \beta_0$ is facet defining for $P(S)$ and for each $e' \in \delta(S)$ there exists a vector $y_{\delta(S)} \in F(S)$ such that $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l = \beta_0$ and $\sum_{l \in L} C^l y_{e'}^l > B(S)$, then the inequality is facet defining for P.*

Proof. Let $S \subset V$ be such that the subgraphs induced by S and $V \setminus S$ are both connected and suppose that inequality $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l \geq \beta_0$ is facet defining for $P(S)$ and for each $e' \in \delta(S)$ there exists a vector $y_{\delta(S)} \in F(S)$ such that $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l = \beta_0$ and $\sum_{l \in L} C^l y_{e'}^l > B(S)$. Assume that $B(S) = b(S)$. The proof for the other case, i.e., $B(S) = b(V \setminus S)$, is similar.

Suppose that all points in P such that $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l = \beta_0$ also satisfy $\overline{\alpha}\lambda + \overline{\beta} y = \overline{\gamma}$. Let $e' \in E \setminus \delta(S)$, $l' \in L$, and $(\lambda, y) \in P$ be such that $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l = \beta_0$. The vector $(\lambda, \overline{y})$ where $\overline{y} = y + v_{e'}^{l'}$ is also in P and satisfies $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l \overline{y}_e^l = \beta_0$. Hence $\overline{\beta}_{e'}^{l'} = 0$.

Define $\lambda_{S \cap W}^{\delta(S)}$ to be the restriction of the vector λ to its entries with edges in $\delta(S)$ and vertices in $S \cap W$. Let $y_{\delta(S)} \in F(S)$ be such that $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l y_e^l = \beta_0$ and $\lambda_{\delta(S)}^{S \cap W}$ be a solution to the system (5.27)-(5.29) with $y_{\delta(S)}$. Let $s' \in W$ and $e' \in E \setminus \delta(S)$. Let $\overline{\lambda} = \sum_{s \in W} \sum_{e \in E \setminus \delta(S)} u_s^e - u_{s'}^{e'} + \sum_{s \in S \cap W} \sum_{e \in \delta(S)} \lambda_s^e u_s^e$ and $\overline{y} = \sum_{l \in L} \sum_{e \in \delta(S)} y_e^l v_e^l + \sum_{l \in L} \sum_{e \in E \setminus \delta(S)} M v_e^l$. The vector $(\overline{\lambda}, \overline{y})$ is in P and satisfies $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l \overline{y}_e^l = \beta_0$. Consider also $(\hat{\lambda}, \overline{y})$ where $\hat{\lambda} = \overline{\lambda} + u_{s'}^{e'}$. This latter vector is also in P (resp. in P') and satisfies $\sum_{l \in L} \sum_{e \in \delta(S)} \beta_e^l \overline{y}_e^l = \beta_0$. Hence $\overline{\alpha}_{s'}^{e'} = 0$.

Let $s' \in W$, $e' \in \delta(S)$ and $y_{\delta(S)} \in F(S)$ be such that $\sum_{l\in L}\sum_{e\in\delta(S)}\beta_e^l y_e^l = \beta_0$ and $\sum_{l\in L} C^l y_{e'}^l > b(S)$. Let $\lambda_{\delta(S)}^{S\cap W}$ be a solution to the system (5.27)-(5.29) with $y_{\delta(S)}$. Consider the vector $(\overline{\lambda}, \overline{y})$ where $\overline{\lambda} = \sum_{s\in W}\sum_{e\in E\setminus\delta(S)} u_e^s + \sum_{s\in S\cap W}\sum_{e\in\delta(S)}\lambda_e^s u_e^s$ and $\overline{y} = \sum_{e\in\delta(S)} y_e^l v_e^l + \sum_{e\in E\setminus\delta(S)} M v_e^l$. This vector is in P and satisfies $\sum_{l\in L}\sum_{e\in\delta(S)}\beta_e^l\overline{y}_e^l = \beta_0$. Now let $\epsilon > 0$ be very small and consider also the vector $(\hat{\lambda}, \overline{y})$ where $\hat{\lambda} = \overline{\lambda} + \epsilon u_{s'}^{e'}$. This vector is also in P since $\sum_{l\in L} C^l y_{e'}^l > b(S)$ and satisfies $\sum_{l\in L}\sum_{e\in\delta(S)}\beta_e^l\overline{y}_e^l = \beta_0$. Thus $\overline{\alpha}_{s'}^{e'} = 0$.

As $F(S) = Proj_{y_{\delta(S)}}(F)$ and as $\sum_{l\in L}\sum_{e\in\delta(S)}\beta_e^l y_e^l \geq \beta_0$ is facet defining for $P(S)$, $\overline{\alpha}\lambda + \overline{\beta}y = \overline{\gamma}$ is a multiple of $\sum_{l\in L}\sum_{e\in\delta(S)}\beta_e^l y_e^l = \beta_0$.

□

For $S \subset V$ and $l \in L$, let

$$Y^l(S) = \sum_{e\in\delta(S)} y_e^l,$$

$$r^l(S) = b(S) - \left\lfloor \frac{b(S)}{C^l} \right\rfloor C^l,$$

and

$$R^l(S) = \lceil B(S) \rceil - \left\lfloor \frac{\lceil B(S) \rceil}{C^l} \right\rfloor C^l.$$

For l_1 and l_2 in L, let

$$f(l_1, l_2) = C^{l_1} - \left\lfloor \frac{C^{l_1}}{C^{l_2}} \right\rfloor C^{l_2}.$$

Now, we modify two well-known families of valid inequalities for the NLP to render them valid for our problem. These inequalities are the cutset inequalities and arc residual capacity inequalities (see e.g. Magnanti et al. [49]). Both inequalities can be generated as Mixed Integer Rounding (MIR) inequalities. Let $X = \{(x_1, x_2) \in \mathbb{R}_+ \times \mathbb{Z} : x_1 + x_2 \geq \mu\}$. The MIR inequality $x_1 \geq (\mu - \lfloor\mu\rfloor)(\lceil\mu\rceil - x_2)$ is valid for X (see Wolsey [69]).

The special cases of the cutset and residual capacity inequalities for the network loading problem under hose uncertainty with a single facility type are

presented and used in Karaşan et al. [44] in order to strengthen the LP relaxation bound. However, they neither give any polyhedral results nor implement a branch-and-cut algorithm.

The set $F(S)$ is an integer knapsack cover set and the polyhedral properties of $P(S)$ are studied in Yaman [70]. A family of valid inequalities, called "lifted rounding inequalities", are generated using sequence independent lifting. These inequalities generalize the cutset inequalities. As they are valid for $P(S)$, they are also valid for P and P'.

Proposition 5.4.4. *For $S \subset V$ and $l^* \in L$ such that $R^{l^*}(S) > 0$, the cutset inequality*

$$\sum_{l \in L: C^l < B(S)} \left(R^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min \left\{ f(l, l^*), R^{l^*}(S) \right\} \right) Y^l(S) + \sum_{l \in L: C^l \geq B(S)} R^{l^*}(S) \left\lceil \frac{B(S)}{C^{l^*}} \right\rceil Y^l(S) \geq R^{l^*}(S) \left\lceil \frac{B(S)}{C^{l^*}} \right\rceil \tag{5.33}$$

is valid for P and P'.

Yaman [70] proves that if $C^1 = 1$, then the cutset inequality (5.33) for $l^* \in L$ such that $R^{l^*}(S) > 0$ is facet defining for $P(S)$. Using Theorem 2, we can state the following proposition.

Proposition 5.4.5. *Let $S \subset V$ be such that the subgraphs induced by S and $V \setminus S$ are both connected and $l^* \in L$ be such that $R^{l^*}(S) > 0$. If $C^1 = 1$, then the cutset inequality (5.33) is facet defining for P.*

Proof. Suppose that $S \subset V$ such that the subgraphs induced by S and $V \setminus S$ are both connected, $l^* \in L$ such that $R^{l^*}(S) > 0$ and $C^1 = 1$. The cutset inequality (5.33) is facet defining for $P(S)$ ([70]). To show that it also defines a facet of P, we need to prove, for each $e \in \delta(S)$, the existence of a vector $y_{\delta(S)} \in F(S)$ which satisfies the inequality (5.33) as equality and $\sum_{l \in L} C^l y_e^l > B(S)$. For edge $e \in \delta(S)$, consider the vector $y_{\delta(S)} = \left\lceil \frac{B(S)}{C^{l^*}} \right\rceil v_e^{l^*}$. This vector satisfies (5.33) as equality, and we have $\sum_{l \in L} C^l y_e^l > B(S)$ since $R^{l^*}(S) > 0$. □

Notice that if $C^2, \ldots, C^{|L|}$ are divisible by C^1, then we can scale the b_s values and the C^l values by dividing with C^1 so that $C^1 = 1$.

If $|L| = 1$ and $R^1(S) > 0$, then the cutset inequality (5.33) is facet defining for P for $S \subset V$ such that the subgraphs induced by S and $V \setminus S$ are both connected.

Next, we generate residual capacity inequalities as MIR inequalities.

Proposition 5.4.6. *Let $e \in E$, $l^* \in L$, and $S \subseteq W$ be such that $r^{l^*}(S) > 0$. The residual capacity inequality*

$$\sum_{l \in L} \left(r^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min \left\{ f(l, l^*), r^{l^*}(S) \right\} \right) y_e^l + \sum_{s \in S} b_s (1 - \lambda_s^e) \geq r^{l^*}(S) \left\lceil \frac{b(S)}{C^{l^*}} \right\rceil \tag{5.34}$$

is valid for P'.

Proof. Inequality $\sum_{s \in S} b_s \lambda_s^e \leq \sum_{l \in L} C^l y_e^l$ is a valid inequality for P'. We substitute $\overline{\lambda}_s^e = 1 - \lambda_s^e$ for $s \in S$ and obtain $b(S) \leq \sum_{l \in L} C^l y_e^l + \sum_{s \in S} b_s \overline{\lambda}_s^e$. Let $l^* \in L$ and divide the last inequality with C^{l^*}. This yields $\frac{b(S)}{C^{l^*}} \leq \sum_{l \in L} \frac{C^l}{C^{l^*}} y_e^l + \sum_{s \in S} \frac{b_s}{C^{l^*}} \overline{\lambda}_s^e$. Let $L_u = \{l \in L : f(l, l^*) > r^{l^*}(S)\}$ and $L_d = L \setminus L_u$. The last inequality implies

$$\frac{b(S)}{C^{l^*}} \leq \sum_{l \in L_d} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor y_e^l + \sum_{l \in L_d} \frac{f(l, l^*)}{C^{l^*}} y_e^l + \sum_{l \in L_u} \left\lceil \frac{C^l}{C^{l^*}} \right\rceil y_e^l + \sum_{s \in S} \frac{b_s}{C^{l^*}} \overline{\lambda}_s^e.$$

Here $\sum_{l \in L_d} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor y_e^l + \sum_{l \in L_u} \left\lceil \frac{C^l}{C^{l^*}} \right\rceil y_e^l$ is integer and $\sum_{l \in L_d} \frac{f(l,l^*)}{C^{l^*}} y_e^l + \sum_{s \in S} \frac{b_s}{C^{l^*}} \overline{\lambda}_s^e$ is a nonnegative real. Hence, the MIR inequality

$$\sum_{l \in L_d} \frac{f(l, l^*)}{C^{l^*}} y_e^l + \sum_{s \in S} \frac{b_s}{C^{l^*}} \overline{\lambda}_s^e \geq \frac{r^{l^*}(S)}{C^{l^*}} \left(\left\lceil \frac{b(S)}{C^{l^*}} \right\rceil - \sum_{l \in L_d} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor y_e^l - \sum_{l \in L_u} \left\lceil \frac{C^l}{C^{l^*}} \right\rceil y_e^l \right)$$

is valid for P'. Multiplying both sides with C^{l^*} and organizing the terms, we obtain

$$\sum_{l \in L_d} \left(f(l, l^*) + r^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \right) y_e^l + \sum_{l \in L_u} r^{l^*}(S) \left\lceil \frac{C^l}{C^{l^*}} \right\rceil y_e^l + \sum_{s \in S} b_s \overline{\lambda}_s^e \geq r^{l^*}(S) \left\lceil \frac{b(S)}{C^{l^*}} \right\rceil$$

which is the same as

$$\sum_{l \in L} \left(r^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min \left\{ f(l, l^*), r^{l^*}(S) \right\} \right) y_e^l + \sum_{s \in S} b_s \overline{\lambda}_s^e \geq r^{l^*}(S) \left\lceil \frac{b(S)}{C^{l^*}} \right\rceil.$$

Substituting $\overline{\lambda}_s^e = 1 - \lambda_s^e$ for $s \in S$ yields inequality (5.34). □

If $|L| = 1$, the residual capacity inequality becomes

$$r^1(S)y_e^1 + \sum_{s \in S} b_s(1 - \lambda_s^e) \geq r^1(S) \left\lceil \frac{b(S)}{C^1} \right\rceil . \tag{5.35}$$

Magnanti et al. [49] prove the following: If $\left\lceil \frac{b(S)}{C^1} \right\rceil \geq 2$, then this inequality defines a facet of P_e'. If $\left\lceil \frac{b(S)}{C^1} \right\rceil = 1$ then the inequality defines a facet of P_e' if $|S| = 1$. Using Theorem 1, we can prove the following:

Proposition 5.4.7. *Let $e \in E$ be such that $\delta(S') \setminus \{e\} \neq \emptyset$ for every $S' \subset V$ such that there exists $(s,t) \in Q$ with $s \in S'$ and $t \in V \setminus S'$. Suppose that $|L| = 1$ and let $S \subseteq W$ be such that $r^1(S) > 0$. The residual capacity inequality (5.35) defines a facet of P' if $\left\lceil \frac{b(S)}{C^1} \right\rceil \geq 2$ or if $\left\lceil \frac{b(S)}{C^1} \right\rceil = 1$ and $|S| = 1$.*

5.5 Branch-and-Cut algorithm

In NLP_{hose}, we have the constraints (5.23), which can be exponential in number. Naturally, it is not reasonable and possible to include all such inequalities at the outset and try to solve the resulting model as it is. Therefore, we will use a branch-and-cut algorithm, which starts with a larger feasible set $\{(\lambda, y) \in \mathbb{R}_+^{|W||E|} \times \mathbb{Z}_+^{|E||L|} : (5.22)\}$ and adds the violated valid inequalities at each iteration. In the rest of this section, we will first explain the separation algorithms we use for the feasibility cuts (5.23) as well as the demand cutset (5.33), and residual capacity (5.34) inequalities. Then, we describe briefly our upper bounding procedure.

5.5.1 Separation of feasibility cuts

Inequalities (5.23) can be separated by solving minimum cut problems. Given a pair $(\bar{\lambda}, \bar{y})$, we construct an auxiliary graph $G_\lambda(st) = (V, E)$ for each origin-destination pair $(s,t) \in Q$ such that the capacity of each edge $e \in E$ is set to be $\bar{\lambda}_s^e + \bar{\lambda}_t^e$. If the capacity of the minimum cut $C(s,t)$ separating s and t is less than 1, then we have a violated inequality (5.23) for (s,t). Otherwise, no inequality (5.23) is violated for (s,t) by the pair $(\bar{\lambda}, \bar{y})$.

5.5.2 Separation of demand cutset inequalities

We have a heuristic separation algorithm for the demand cutset inequalities (5.33). For each commodity $(s,t) \in Q$, we use the cut $C(s,t)$ for which a feasibility cut (5.22) is violated. If the pair $(\bar{\lambda}, \bar{y})$ also violates a demand cutset inequality for $C(s,t)$, then we add the corresponding cut to the problem.

5.5.3 Separation of residual capacity inequalities

We do not know any polynomial time algorithm to separate the residual capacity inequalities (5.34). But we can separate a relaxed version of these inequalities in polynomial time.

Let $e \in E$, $l^* \in L$, and $S(e,l^*) \subseteq W$. Define the relaxed residual capacity inequality as

$$\sum_{l \in L} \left(C^l - \left(C^{l^*} - r^{l^*}(S(e,l^*)) \right) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \right) y_e^l + \sum_{s \in S(e,l^*)} b_s(1 - \lambda_s^e) \geq r^{l^*}(S(e,l^*)) \left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil. \quad (5.36)$$

This inequality is valid for P' as it is implied by inequality (5.34). Moreover, it is a MIR inequality.

For a given edge $e \in E$, a facility type $l^* \in L$, and a pair $(\overline{\lambda}^e, \overline{y}_e)$, finding a violated relaxed residual capacity inequality or showing that there is no such inequality is equivalent to solving the problem

$$\min_{S(e,l^*) \subseteq W} \left\{ \sum_{s \in S(e,l^*)} b_s(1 - \overline{\lambda}_s^e) - r^{l^*}(S(e,l^*)) \left(\left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil - \sum_{l \in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l \right) \right\}$$

with the optimal value $\phi(e,l^*)$. If $\sum_{l \in L} \left(C^l - C^{l^*} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \right) \overline{y}_e^l + \phi(e,l^*) \geq 0$ then all relaxed residual capacity inequalities for $e \in E$ and $l^* \in L$ are satisfied by $(\overline{\lambda}^e, \overline{y}_e)$. Otherwise, we have a violated relaxed residual capacity inequality defined by a minimizing set $S(e,l^*)$.

Claim 1. *The relaxed residual capacity inequality cannot be violated unless* $\left\lfloor \frac{b(S(e,l^*))}{C^{l^*}} \right\rfloor < \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l < \left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil$.

Proof. Since (5.34) is the MIR inequality for the relaxation $\sum_{s\in S(e,l^*)}(1-\lambda_s^e)b_s + \sum_{l\in L} C^l y_e^l \geq b(S(e,l^*))$, it would be dominated by $\sum_{s\in S(e,l^*)}(1-\lambda_s^e)b_s \geq 0$ and $\sum_{s\in S(e,l^*)}(1-\lambda_s^e)b_s + \sum_{l\in L} C^l y_e^l \geq b(S(e,l^*))$ when $\sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor y_e^l \geq \left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil$ or $\sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor y_e^l \leq \left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil - 1$. □

Then, using the arguments in Atamtürk and Rajan [5], we can show that the relaxed residual capacity inequalities can be separated in the following way: Let

$$S(e,l^*) = \left\{ s \in W : \overline{\lambda}_s^e > \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l - \left\lfloor \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l \right\rfloor \right\}$$

and

$$\Psi(S(e,l^*)) = \sum_{s\in S(e,l^*)} b_s(1-\bar{\lambda}_s^e) - r^{l^*}(S(e,l^*)) \left(\left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil - \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}_e^l \right).$$

If $\left\lfloor \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l \right\rfloor < \frac{b(S(e,l^*))}{C^{l^*}} < \left\lceil \sum_{l\in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \overline{y}_e^l \right\rceil$ and $\sum_{l\in L} \left(C^l - C^{l^*} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \right) \overline{y}_e^l + \Psi(S(e,l^*)) < 0$, then the relaxed residual capacity inequality for edge $e \in E$, $l^* \in L$, and the set $S(e,l^*)$ is violated. Otherwise, no inequality (5.36) for this choice of edge and facility type is violated. Hence, separation of the relaxed residual capacity inequalities can be done in $O(|E||L||W|)$ time. Note that we solve the separation problem for the relaxed residual capacity inequalities (5.36) but add the stronger inequalities (5.34) in case of a violation. Another alternative is to use a hybrid separation method where, for each edge e and facility type l^*, we check if any strong inequality (5.34) is violated for the set $S(e,l^*)$. We have implemented both methods and observed that the former method is as efficient as the latter one. Hence, we use the separation algorithm displayed in Algorithm 5.1 for the residual capacity inequalities (5.34).

Algorithm 5.1 Residual capacity inequality separation

for all edge $e \in E$ **do**
  for all facility type $l^* \in L$ **do**
    $\bar{Y}_e^{l^*} := \sum_{l \in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}_e^l$
    for all terminal $s \in W$ **do**
      if $\bar{\lambda}_s^e > \bar{Y}_e^{l^*} - \left\lfloor \bar{Y}_e^{l^*} \right\rfloor$ **then**
        $S(e, l^*) := S(e, l^*) \cup \{s\}$
    if $\left\lfloor \bar{Y}_e^{l^*} \right\rfloor < \frac{b(S(e,l^*))}{C^{l^*}} < \left\lceil \bar{Y}_e^{l^*} \right\rceil$ and $\sum_{l \in L} \left(C^l - C^{l^*} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \right) \bar{y}_e^l +$ $\sum_{s \in S(e,l^*)} b_s(1 - \bar{\lambda}_s^e) - r^{l^*}(S(e, l^*)) \left(\left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil - \sum_{l \in L} \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}_e^l \right) < 0$ **then**
      Add the violated residual capacity inequality
      $\sum_{l \in L} (r^{l^*}(S(e, l^*)) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min\{f(l, l^*), r^{l^*}(S(e, l^*))\}) y_e^l +$ $\sum_{s \in S(e,l^*)} b_s(1 - \lambda_s^e) \geq r^{l^*}(S(e, l^*)) \left\lceil \frac{b(S(e,l^*))}{C^{l^*}} \right\rceil$

5.5.4 Heuristics

Given the difficulty of the problem, we expect it to be useful to incorporate some approximation heuristics into our B&C algorithm. These algorithms yield easy-to-compute upper bounds, useful especially for the large instances that we could not solve to optimality within our time limits.

We apply a simple rounding heuristic to get upper bounds on the optimal solution. So, at each node of the B&C tree, if we cannot find any violated inequality then we have a feasible solution for the LP relaxation of the NLP_{hose} problem. Let $(\bar{\lambda}, \bar{y})$ be the current solution. We simply generate a feasible solution $(\bar{\lambda}, \hat{y})$ such that $\hat{y}_e^l = \lceil \bar{y}_e^l \rceil$ for all $e \in E$ and $l \in L$. Bienstock et al. [16] also use a similar method and mention that it is efficient.

We have also tried a more advanced heuristic, which is an adaptation of the approximation algorithm developed in Gupta et al. [30] for designing Virtual Private Networks with continuous capacity reservation. However, the results were not promising for our problem and the simple rounding heuristic was superior. Thus, we use only the rounding heuristic in our B&C algorithm.

5.6 Experimental results

In this section we report the results of a computational study for NLP_{hose} with single facility and two facilities. Let NLP_{GD}^{hose} be the NLP_{GD} model for the hose uncertainty definition, which we solve using Cplex. Then, we compare the performance of the B&C algorithm with that of Cplex on instances from the network design literature. Most of these instances, namely *polska, dfn, newyork, france, janos, atlanta, tai, nobel-eu, pioro, gui39, cost266, norway,* and *sun*, are from the SNDLIB web site ([35]) whereas the remaining 7 are the ones used in Chapter 3 for the VPN design problem. For the SNDLIB instances the average pairwise demand estimates d_{st} are available. Hence, in order to generate the corresponding hose polyhedron, we let the bandwidth of each terminal node to be the total demand incident to it. In other words, we let $b_s = \sum_{t \in W \setminus \{s\}} (d_{st} + d_{ts})$ for $s \in W$ be the terminal node bandwidths.

We have used AMPL to model the single and two facility variants of NLP_{GD}^{hose} as well as Cplex 9.1 MIP solver to solve them. The B&C algorithm is implemented in C using MINTO (Mixed INTeger Optimizer [59]) and Cplex 9.1 as LP solver. We have set a two-hour time limit both for AMPL and MINTO. The branching rule for the B&C algorithm is to choose the integer variable with fractional part closest to 0.5 to create two subproblems at each node of the B&C tree, where the current solution is not integral. Moreover, node selection is done using the best-bound search of the B&C tree. We present our test results for single- and two-facility cases in Sections 5.6.1 and 5.6.2, respectively. The entries in the tables contain the following information:

- the instance characteristics, i.e., the name of the instance as well as the numbers of nodes, edges, and terminals[1],
- facility capacities, i.e., C for the single facility case, C^1 and C^2 for the two facility case,
- the best upper bound z_{cp} and CPU time t_{cp} of Cplex,

[1]Then the number of commodities is $|W|(|W|-1)$.

- the best upper bound $z_{B\&C}$ and CPU time $t_{B\&C}$ of the B&C algorithm,
- the number of B&C nodes for Cplex $\#_{cp}$,
- the number of B&C nodes for our algorithm $\#_{B\&C}$,
- the gap at termination for Cplex g_{cp},
- the gap at termination for B&C $g_{B\&C}$,

where gap at termination is the gap between the best upper and lower bounds at termination for each method. Given the difficulty of the problem, it is not surprising that some of the instances are not solved to optimality within two hours time limit. We indicate such cases with a * in the corresponding z column. Moreover, the gaps at termination (g) for them are provided under the corresponding time (t) columns in parenthesis. Besides, we mark those cases for which solving the NLP_{GD}^{hose} or NLP_{hose} models did not yield even a feasible solution within the time limit with NoI whereas mem indicates a termination before the end of two-hour time limit due to insufficient memory allocation. We do not provide any gap at termination for such cases and mark them as "–" in all the result tables. Finally, note that all solution times are in CPU seconds.

5.6.1 Single facility NLP_{hose}

For the single facility case, we assume that there is only one type of facility available with a capacity of C units. Then, the demand cutset inequalities (5.33) reduce to

$$Y^1(S) \geq \left\lceil \frac{B(S)}{C} \right\rceil \quad \forall S \subset V, \tag{5.37}$$

which ensures that the total capacity across a cut is sufficient to support the total demand between all terminal pairs whose end points are on different shores of the cut. Moreover, the residual capacity inequalities are

$$\sum_{s \in S \cap W} \frac{b_s}{C}(1 - \lambda_s^e) \geq \left(\frac{b(S)}{C} - \left\lfloor \frac{b(S)}{C} \right\rfloor \right) \left(\left\lceil \frac{b(S)}{C} \right\rceil - y_e \right) \quad \forall S \subset V,\ e \in E \tag{5.38}$$

whose separation can be done in $O(|W||E|)$ time using Algorithm 5.1 ([5]). Notice that the inequalities (5.34) and (5.36) are identical for the single facility case. Hence we implement an exact separation algorithm for the residual capacity inequalities (5.34).

The first set of results are shown in Table 5.1 where we compare our B&C algorithm with solving the single facility NLP^{hose}_{GD} using Cplex. At this stage, we include the demand cutset inequalities (5.37) and the arc residual capacity inequalities (5.38) together with the feasibility cuts (5.23) in the B&C algorithm. However, we will also provide an analysis of the effect of each family of valid inequalities on the solution quality later in this section.

Table 5.1: Results for the single facility problem.

name	(V,E,W,C)	z_{cp}	$t_{cp}(g\%)$	$z_{B\&C}$	$t_{B\&C}(g\%)$
metro	(11,42,5,24)	818	31	818	19.97
nsf1b	(14,21,10,24)	97,100	103.5	97,100	3.68
at-cep1	(15,22,6,24)	51,745	2.84	51,745	0.39
pacbell	(15,21,7,24)	11,390	12.46	11,390	1.76
bhv6c	(27,39,15,24)	840,251	110.56	$840,251^*$	(1.22%)
bhvdc	(29,36,13,24)	1.1e+6	1098.44	1.1e+6	3.6
pdh	(11,34,6,480)	1.1e+6	11.39	1.1e+6	0.22
polska	(12,18,12,155)	$44,253^*$	(1.16%)	$44,287^*$	(0.42%)
polska	(12,18,12,1000)	7478	3713	7478	3094.19
dfn	(11,47,11,155)	$52,380^*$	(6.34%)	$52,416^*$	(3.74%)
newyork	(16,49,16,1000)	$1.499e+6^*$	(56.74%)	$1.497e+6^*$	(45.4%)
france	(25,45,14,2500)	$17,400^*$	(3.61%)	$17,400^*$	(5.45%)
ny-cep2	(16,49,9,24)	7079.2^*	(3.8%)	7094.4^*	(2.52%)
atlanta	(15,22,15,1000)	$4.7e+8^*$	(0.2%)	$4.72e+8^*$	(0.54%)
tai	(24,51,19,504k)	$1.6e+12^*$	(99.7%)	$3.6e+7^*$	(19.94%)
janos	(26,42,26,64)	$1.29e+9^*$	(99.7%)	$2.7e+6$	884.31
nobel-eu	(28,41,28,20)	$1.47e+10^*$	(99.7%)	$4.3e+6^*$	(2.01%)
sun	(27,51,24,40)	$6.3e+10^*$	(99.7%)	1533.34^*	(20.5%)

Out of 18 instances, Cplex and B&C could both solve 8 to optimality in 2 hours. In *bhv6c*, B&C could find the optimal solution with the rounding method but since it could not improve the lower bound, it kept on branching till the end of the time limit. Moreover, even though Cplex gives better upper bounds than

B&C in *dfn, ny-cep2*, and *atlanta*, the gaps at termination are better for the B&C algorithm in the first two of these instances. On the other hand, B&C is clearly superior for the instances *newyork, tai, janos, nobel-eu,* and *sun*. Actually, the most important point to be highlighted is the significant degradation in the performance of Cplex relative to the B&C algorithm as the network size increases. The last four instances on Table 5.1 are very good examples of this behavior. Except *tai*, all of the nodes are demand nodes in these instances, i.e., $W = V$, and we observe that among such cases only in *dfn* and *atlanta* Cplex has performed slightly better than B&C. Actually the upper bound of Cplex is just 0.07% and 0.2% smaller than the one of B&C in *dfn* and *atlanta*, respectively. On the other hand, the upper bounds we obtain with B&C are 100% better than the bounds with Cplex in *tai, janos, nobel-eu,* and *sun*. Finally, a comparison of the gaps at termination shows that the B&C algorithm is clearly superior in 8 of the 11 instances with much lower gaps for the *tai*, *nobel-eu*, and *sun* in addition to the zero gap for the *janos* instance.

Next, we provide further information on the B&C tree statistics in Table 5.2, where we provide the root relaxation solution times t_{cp}^{root} and $t_{B\&C}^{root}$, the numbers of B&C nodes $\#_{cp}$ and $\#_{B\&C}$ as well as the root relaxation gaps gap_{cp} and $gap_{B\&C}$ for Cplex and our B&C algorithm, respectively. Root relaxation gap is defined as the gap between the best upper bound at termination and the lower bound at the root. We mark those instances, which could not be solved to optimality in 2 hours, with a $^+$ sign under the *gap* columns.

Table 5.2 shows that although the root solution times are shorter for Cplex, the root relaxation gaps of our B&C algorithm are significantly smaller than the ones of Cplex. Given that the B&C results belong to the case where both the residual capacity and demand cutset inequalities are used, we are confident to say that these cuts are quite useful for improving the root relaxation bounds.

An additional analysis we have performed is the comparison of the individual and joint influence of the two types of cuts on the root relaxation solution qualities and the total solution times when they are used together with the feasibility cuts. We investigate the following four cases:

Table 5.2: Further comparison of B&C statistics for the single facility case.

name	t_{cp}^{root}	$\#_{cp}$	gap_{cp}	$t_{B\&C}^{root}$	$\#_{B\&C}$	$gap_{B\&C}$
metro	0.15	1092	8.83%	0.11	913	3.02%
nsf1b	0.21	504	1.56%	0.89	21	0.34%
at-cep1	0.03	70	4.42%	0.26	5	0.5%
pacbell	0.09	304	3.54%	0.34	16	0.37%
bhv6c	0.14	4949	2.25%	0.9	37,702	$0\%^{+}$
bhvdc	0.26	590	1.18%	3.56	3	0.02%
pdh	0.21	78	9.06%	0.22	1	0.4%
polska(155)	0.18	10,793	$2.14\%^{+}$	1.73	46,133	$0.48\%^{+}$
polska(1000)	0.18	8056	$10.24\%^{+}$	1.44	18,719	$2.56\%^{+}$
dfn	1.28	1159	$10.12\%^{+}$	1.91	6624	$4.44\%^{+}$
newyork	9.11	60	$69.5\%^{+}$	44.36	21	$48.48\%^{+}$
france	1.26	525	$8.55\%^{+}$	23.25	304	$6.03\%^{+}$
ny-cep2	0.29	2741	$5.5\%^{+}$	0.94	7998	$2.76\%^{+}$
atlanta	0.54	9594	$0.95\%^{+}$	3.04	19,135	$0.61\%^{+}$
tai	108.66	52	$99.99\%^{+}$	1286.41	13	$20.13\%^{+}$
janos	19.81	30	$99.79\%^{+}$	551.02	3	$0.36\%^{+}$
nobel-eu	28.2	25	$99.97\%^{+}$	1934.01	8	$2.06\%^{+}$
sun	92.35	1	$99.97\%^{+}$	2166.63	6	$20.8\%^{+}$

- *F*: use only the feasibility cuts;
- *F+D*: use feasibility cuts and demand cutset inequalities together;
- *F+R*: use feasibility cuts and residual capacity inequalities together;
- *all*: use all three types of cuts together.

For each of these four settings, we report the percentage gap between the optimal value and the lower bound at the root node under the *gap* columns as well as the corresponding solution times under the t columns in Table 5.3. We have considered 6 instances, which were solved to optimality in relatively shorter times. Based on the results for these six instances we can say that the impact of demand cutset inequalities both on root gaps and solution times is significant. Moreover, the residual capacity inequalities have also yield reasonable improvements in the root gaps. Although adding residual capacity inequalities together with demand cutset inequalities does not improve the root gaps, it improves the solution times. The root gaps and solution times are improved by 86% and 84.72%, respectively on the average as we use all cuts in our B&C algorithm.

Table 5.3: Results with different cuts.

name	t_F	gap_F	t_{F+D}	gap_{F+D}	t_{F+R}	gap_{F+R}	t_{all}	gap_{all}
metro	37.34	8.83%	27.72	3.00%	46.32	7.89%	19.97	3.02%
nsf1b	223.53	1.56%	4.34	0.34%	93.79	1.04%	3.68	0.34%
at-cep1	4.56	4.44%	0.47	0.50%	3.69	2.69%	0.39	0.50%
pacbell	68.10	3.54%	3.65	0.37%	73.59	2.87%	1.76	0.37%
bhvdc	7114.04	1.18%	15.83	0.07%	767.66	0.63%	3.60	0.02%
pdh	0.87	9.06%	0.29	0.00%	1.35	4.55%	0.22	0.40%

Finally, Table 5.4 shows the change in both the solution times and the optimal capacity installation cost when demand uncertainty is added to the model. We believe this gives a better sense of how the original NLP is affected on the average by the inclusion of demand uncertainty. The Δ_z column shows the changes in the total reservation costs as we shift from the deterministic NLP to its robust counterpart and it is observed that the average increase is 17.62%. This figure

can be interpreted as the price of robustness the network service provider should be ready to pay. On the other hand, we observe some interesting results for the solution times. When we solve the deterministic NLP and NLP_{GD}^{hose} with Cplex, we see that the solution times for the polyhedral case are much shorter for the instances *pacbell, nsf1b, bhvdc,* and *bhv6c*. We believe, this is due to the nice structure of NLP_{GD} we have obtained in Proposition 5.1.1. Obviously, this is a very important and particular observation we can make on our models. Normally, the robust optimization problems are expected to be more difficult than their deterministic counterparts, i.e., the price for the flexibility of the robust models is paid by means of additional computational efforts together with an aggravation in the optimal objective value generally. But for our case, the robust problem is even easier to solve and hence there is no technical reason why one would avoid the advantages of using the hose model.

Table 5.4: Deterministic vs polyhedral single facility NLP.

name	t_{det}	z_{det}	t_{hose}	z_{hose}	Δ_z
metro	3.16	699	31	818	14.5%
nsf1b	785.74	70,100	103.04	97,100	27.8%
at-cep1	0.75	43,110	2.84	51,745	16.7%
pacbell	53.44	8660	12.46	11,390	24%
bhv6c	558.59	783,376	110.6	840,251	6.8%
bhvdc	3559.4	851,889	1098.44	1,095,622	22.2%
pdh	3.17	1,012,673	11.39	1,142,081	11.43%

5.6.2 Two facility NLP_{hose}

In the two facility case, we distinguish between the facility types according to their transmission capacities and refer to them as low capacity facility (LCF) and high capacity facility (HCF). We assume that the capacities of each LCF and HCF are C^1 and C^2 units, respectively. Naturally, the cost of installing each facility is different and economies of scale prevails, i.e., the cost of $\left\lceil \frac{C^2}{C^1} \right\rceil$ LCF's is more than the cost of 1 HCF. Then, for $S \subset V$, the demand cutset inequalities (5.33) reduce to the following inequalities:

- *The LCF case*, i.e., $l^* = 1$, where the resulting inequalities are in either of the following forms:

$$\begin{cases} R^1(S)Y^1(S) + (R^1(S)\left\lfloor \frac{C^2}{C^1} \right\rfloor + \\ \qquad \min\{f(2,1), R^1(S)\})Y^2(S) \geq R^1(S)\left\lceil \frac{B(S)}{C^1} \right\rceil & :C^1, C^2 < B(S) \\ Y^1(S) + Y^2(S) \geq 1 & :C^1, C^2 \geq B(S) \\ Y^1(S) + \left\lceil \frac{B(S)}{C^1} \right\rceil Y^2(S) \geq \left\lceil \frac{B(S)}{C^1} \right\rceil & :C^1 < B(S) \\ & \text{and } C^2 \geq B(S). \end{cases}$$

- *The HCF case*, i.e., $l^* = 2$ where we can have:

$$\begin{cases} \min\{C^1, R^2(S)\}Y^1(S) + R^2(S)Y^2(S) \geq R^2(S)\left\lceil \frac{B(S)}{C^2} \right\rceil & : C^1, C^2 < B(S) \\ Y^1(S) + Y^2(S) \geq 1 & : C^1, C^2 \geq B(S) \\ C^1 Y^1(S) + \lceil B(S) \rceil Y^2(S) \geq \lceil B(S) \rceil & : C^1 < B(S) \\ & \text{and } C^2 \geq B(S). \end{cases}$$

On the other hand, the two types of residual capacity inequalities (5.34) for each edge $e \in E$ and set $S \subset V$ are

$$\begin{cases} r^1(S)y_e^1 + \left(r^1(S)\left\lfloor \frac{C^2}{C^1} \right\rfloor + \min\{f(2,1), r^1(S)\}\right) y_e^2 - \\ \qquad \sum_{s \in S \cap W} b_s \lambda_s^e \geq r^1(S)\left\lceil \frac{b(S)}{C^1} \right\rceil - b(S) & \text{for} \quad l^* = 1 \\ \min\{C^1, r^2(S)\}y_e^1 + r^2(S)y_e^2 - \sum_{s \in S \cap W} b_s \lambda_s^e \geq r^2(S)\left\lceil \frac{b(S)}{C^2} \right\rceil - b(S) & \text{for} \quad l^* = 2. \end{cases}$$

Notice that the number of residual capacity and demand cutset inequalities are doubled as we move from the single facility to the two facility case. Hence, the LP models we solve at each iteration of the B&C algorithm can get large more quickly. Thereof, we have tried the following schemes for adding violated cuts:

- *HA*: add only HCF type inequalities in all nodes of the B&C tree;
- *HR*: add only HCF type inequalities only at the root node;
- *GHA*: add HCF type inequalities in all nodes of the B&C tree gradually, i.e., add a violated HCF residual capacity inequality only if no HCF demand cutset inequality is violated;

- *GHR*: add HCF type inequalities only at the root node of the B&C tree gradually, i.e., add a violated HCF residual capacity inequality only if no HCF demand cutset inequality is violated;
- *GAR*: add all valid inequalities at the root node gradually, i.e. add violated LCF and HCF residual inequalities only if no LCF or HCF demand cutset inequality is violated.

These five different alternatives yield very similar results where we observe a slight superiority of *GHR* and *GHA*. Thus, we will provide the results for *GHA*, which we observe to be a little better.

An initial comparison of our B&C algorithm and Cplex can be made using the results in Table 5.5. The performance of our B&C algorithm is better than that of Cplex especially for the large instances where all nodes are demand nodes just like the single facility case. This is quite obvious especially for the 4 instances *tai, nobel-eu, pioro*, and *cost266* since the MIP solver could not find even a feasible solution by solving the two facility NLP_{GD}^{hose} problem in 2 hours whereas the B&C algorithm successfully produced some upper bounds. Especially, the upper bounds for *nobel-eu* and *pioro* are quite promising. Moreover, the NLP_{GD}^{hose} problem could not be solved for *newyork* instance due to insufficient memory. On the other hand, for those instances that could be solved optimally within the 2 hours time limit, we observe that B&C has worse solution times only for *pacbell* and *bhv6c*. In two cases, i.e., *norway* and *gui*, we could not find any upper bound with either of the methods[2]. On the other hand, the B&C algorithm is better in 6 of the remaining 9 instances with much lower gaps for *dfn, tai, nobel-eu, pioro*, and *cost266*.

We provide additional information about the B&C tree statistics in Table 5.6, where we provide the root relaxation solution times t_{cp}^{root} and $t_{B\&C}^{root}$, the number of B&C nodes $\#_{cp}$ and $\#_{B\&C}$ as well as the root relaxation gaps gap_{cp} and $gap_{B\&C}$ for Cplex and our B&C algorithm, respectively. We again mark those instances, which could not be solved to optimality in 2 hours, with a $^+$ sign under the *gap*

[2] As we show in Figure 5.2, we could get bounds for these two instances in other cut schemes *HA* and *HR*.

Table 5.5: Gradual inclusion of HCF valid inequalities at each iteration, i.e., the *GHA* setting.

name	(V, E, W, C^1, C^2)	z_{cp}	t_{cp}(g%)	$z_{B\&C}$	$t_{B\&C}$(g%)
metro	$(11, 42, 5, 1, 24)$	772	2.82	772	1.12
nsf1b	$(14, 21, 10, 1, 24)$	96,035	2.82	96,035	1.86
at-cep1	$(15, 22, 6, 1, 24)$	50,968	0.88	50,968	0.49
pacbell	$(15, 21, 7, 1, 24)$	11,177	4.33	11,177	54.14
bhv6c	$(27, 39, 15, 1, 24)$	826,136	1.94	829,548*	(0.5%)
bhvdc	$(29, 36, 13, 1, 24)$	1.09e+6	30.75	1.09e+6	16.12
pdh	$(11, 34, 6, 30, 480)$	8.18e+6	7.17	8.18e+6	3.10
polska	$(12, 18, 12, 155, 622)$	34,006*	(2.18%)	34,348*	(3.11%)
dfn	$(11, 47, 11, 155, 622)$	44,352*	(21.37%)	45,316*	(12.43%)
newyork	$(16, 49, 16, 1k, 4k)$	NoI	mem(–)	1.6e+6*	(70.4%)
atlanta	$(15, 22, 15, 1k, 4k)$	1.37e+8*	(1.15%)	1.38e+8*	(1.44%)
tai	$(24, 51, 19, 504k, 1008k)$	NoI	–	3.06e+7*	(10.48%)
nobel-eu	$(28, 41, 28, 20, 40)$	NoI	–	2.66e+6*	(1.6%)
pioro	$(40, 89, 40, 155, 622)$	NoI	–	1.38e+6	(2.47%)
norway	$(27, 51, 27, 1k, 4k)$	NoI	–	NoI	–
cost266	$(37, 57, 37, 7560, 30240)$	NoI	–	2.42e+7*	(29.75%)
gui39	$(39, 86, 39, 160, 320)$	NoI	–	NoI	–

columns. Moreover, we indicate the instances for which the root relaxation could not be solved within the time limit as 2hrs under the t columns.

Table 5.6: Additional information for the two-facility problem.

name	t_{cp}^{root}	$\#_{cp}$	gap_{cp}	$t_{B\&C}^{root}$	$\#_{B\&C}$	$gap_{B\&C}$
metro	0.14	45	3.39%	0.12	55	0.96%
nsf1b	0.21	33	0.47%	0.86	9	0.004%
at-cep1	0.04	9	2.96%	0.23	11	0.007%
pacbell	0.05	100	1.70%	0.38	1301	0.37%
bhv6c	0.15	16	0.58%	1.19	36,108	$0.99\%^{+}$
bhvdc	0.30	16	0.53%	3.37	43	0.01%
pdh	0.21	49	8.79%	0.21	87	1.43%
polska	0.23	1795	4.94%	1.04	50,403	$3.63\%^{+}$
dfn	0.46	896	27.51%	0.86	21,736	$15.76\%^{+}$
newyork	90.44	2	–	105.77	30	$65.63\%^{+}$
atlanta	0.55	4326	3.85%	2.98	14,598	$1.78\%^{+}$
tai	86.84	127	–	45.87	8702	$10.73\%^{+}$
nobel-eu	34.69	50	–	152.21	690	$3.44\%^{+}$
pioro	1710.71	2	–	14400	2	$2.47\%^{+}$
norway	130.30	2	–	2hrs	1	–
cost266	314.8	2	–	14400	2	$29.78\%^{+}$
gui39	840.86	2	–	2hrs	1	–

Table 5.6 shows that the superiority of our B&C algorithm for the single facility case is still in action for the two facility case. We again see that the HCF type demand cutset and residual capacity inequalities are effective in strengthening the LP relaxations and hence in improving the lower bounds at the root node.

In addition, we compare the performance of the five settings in terms of the gaps at termination as shown in Figure 5.2. The instances for which the B&C algorithm could not find a feasible solution within the 2 hours time limit are assigned a 105% gap. Furthermore, we leave the *bhv6c* instance out of this analysis since all schemes stopped with the same gap. Consequently, we see that the average gaps at termination for these 11 instances are 32.6%, 38.5%, 31.1%, 31.2%, and 56.9% for *HA, HR, GHA, GHR,* and *GAR*, respectively. On the other hand, the average number of nodes in the B&C tree for these five settings are 13968, 11769, 7869, 8903, and 8629. An important point to note here is that

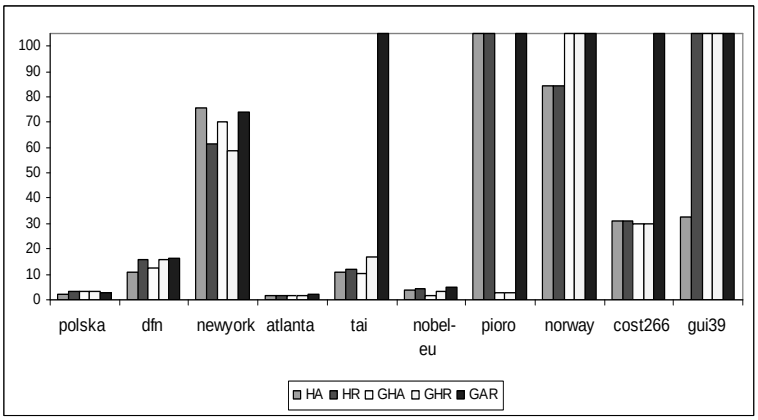

Figure 5.2: Percent gaps at termination for each scheme.

the number of nodes is 1 for those instances terminated with no feasible solution. So, even though the highest number of such cases are observed for *GAR*, the size of the B&C tree is smaller for *GHA* on the average.

A final analysis in Table 5.7 is about the price of robustness measured in terms of the final design cost and solution time for the two facility case. The average increase in the optimal reservation costs of the 6 instances for which gaps could be calculated is 18.86%. *pdh* instance is not included here since the deterministic case could not be solved within 2 hours although the solution time for its robust counterpart is just 3 seconds. Similarly, the difference between the solution times is also obvious for the instances *nsf1b, bhvdc,* and *bhv6c*. Hence, we can say that the nice structure we have obtained in Proposition 5.1.1, is again in effect for the two facility problem.

5.7 Concluding remarks

In this chapter we have discussed the Network Loading Problem where the pair-wise traffic demands are not assumed to be known in advance. We have used a polyhedral definition of traffic demands and sought to design a network which is capable to support infinitely many demand realizations. While the NLP with

Table 5.7: Deterministic vs polyhedral two facility NLP.

name	t_{det}	z_{det}	t_{pol}	z_{pol}	Δ_z
metro	0.86	662	1.12	772	14.2%
nsf1b	7.14	69,216	1.86	96,035	27.9%
at-cep1	0.74	42,578	0.49	50,968	16.5%
pacbell	2.47	8354	4.33	11,177	25.3%
bhv6c	16.4	770,185	1.94	826,136	6.8%
bhvdc	204.49	843,206	16.12	1,088,422	22.5%

known (deterministic) demands is well-studied, we are not aware of any other attempt to study the NLP under polyhedral demand uncertainty with the exception of an earlier reference by Karaşan et al. [44] where uncertainty was incorporated into the design of fiber optic networks with an emphasis on modeling rather than on a detailed polyhedral analysis and branch-and-cut. An astonishing feature of the present formulation for NLP with polyhedral uncertainty is that we avoid the use of metric inequalities due to a decomposition property obtained from a projection on the design components. A similar projection is used in Mirchandani [52] both for single- and multi-commodity NLP with deterministic demands where all extreme rays of the related projection cone for the single-commodity case were characterized. On the other hand, only necessary conditions were obtained for the multi-commodity variant. The difficulty of the latter problem is due to the coupling bundle constraints, which prevent the decomposition of the problem into single commodity subproblems. However, we by-pass that difficulty using the observation that the existence of a multi-commodity flow can be certified by checking the existence of many single commodity flows, i.e., the projection problem can be decomposed into many smaller single-commodity problems for which the results of [52] remain valid. This result considerably simplifies the formulations, and opens the way to a thorough polyhedral analysis of NLP under a specific hose uncertainty definition that is well accepted in the telecommunications literature. Based on the polyhedral analysis we have developed a branch-and-cut algorithm along with a simple heuristic for computing upper bounds, and used it to solve several well-known network design instances.

The studies on the polyhedral properties of the deterministic NLP are limited to the case of at most three facility types where the capacity of a facility is an integer multiple of the capacity of the smaller facility, e.g., if there are three facility types, their capacities are 1, λ, and λC where λ and C are positive integers different from 1. The second main contribution of the present study is that valid inequalities we have presented are valid for an arbitrary number of facilities and arbitrary capacity structures.

Finally, our computational results have revealed that projecting out the flow variables and the use of branch-and-cut algorithm is quite effective for both single and two-facility problem types. An important question is whether similar developments can be expected for uncertainty polyhedron descriptions other than the hose model. We address this problem as the direction of our future studies.

Chapter 6

Conclusion

The telecommunications industry is the rising star of the information age. Internet has led to an evolutionary leap in the field. It has triggered an interest in the use of telecommunication networks for business. To keep up with the increasing demand, quite a few companies offering similar services have entered the market. So, there is a fierce competition and network service providers must develop a clever strategy to have an edge over the others. Naturally, one way of differing from the competitors is to proliferate the product range. This requires more effort on the research-and-development activities. Besides, the price and the quality of the services are also two important competitive advantages. If a company cannot control its costs, then it is unlikely to offer quality products at competitive prices. As a result, the management of network resources by means of optimization tools have become an attractive research area for the operations research community.

Operations research efforts should keep abreast of the changes in the nature of the business, which give rise to new types of service requests, in order to produce practically useful solutions. Existing body of knowledge on network management should be improved to accommodate the dynamic structure of the industry. Therefore, robust network design is becoming more and more important. The motivation of our study is to incorporate the uncertainty in demand estimations into the three eminent problems: Virtual Private Network design, optimal oblivious OSPF routing, and Network Loading Problem. We provide link capacity and

routing configurations which remain operational for a set of demand realizations.

In the rest of this chapter, we will first summarize our main contributions in Section 6.1. Then, we will conclude with a discussion of some future research directions, which we believe to deserve further attention, in Section 6.2.

6.1 Main contributions

As a result of a circumstantial review of the available literature, we have determined three problems. Our main concern is to contribute to the current literature by taking the robustness dimension into consideration. We will summarize our main contributions for each problem in the remaining parts of this section.

6.1.1 Virtual Private Network design

Virtual Private Networks (VPNs) are offered as cost effective alternatives to the devoted private networks. They operate over publicly available larger networks or the private network of the service provider and hence offer several advantages to its users as we have discussed in Section 2.2. Although there are some efforts in the literature to build flexible designs by using the hose model for demand definition, there is no compact formulation of the problem. Several researchers have tried to deal with the semi-infinite nature of the problem by using approximation algorithms or iterative methods, which consider the vertices of the demand polyhedron one by one to find the optimal design. Moreover, only the hose model for representing uncertainty is used in all but one of the available studies. Ben-Ameur and Kerivin [12] mention polyhedral demands but use the hose model in their numerical experiments. So, there is a need to investigate alternative demand polyhedra.

Based on these observations, we can list our main contributions for the VPN design problem as follows:

- We provide an explicit and compact mixed integer programming model for the Asym-G and Sym-G problems. Prior to our work, such models were not available.

- We introduce a new problem Rob-G where we consider a less conservative robust approach in the style of Bertsimas and Sim [14, 15]. We also give a compact formulation and show that unlike its nominal counterpart, Rob-G is actually NP-hard by a polynomial time reduction from the 3-SAT problem.

- Although we can solve our compact formulations for medium-to-large instances using the commercial MIP solvers, we implement a branch-and-price and cutting plane algorithm to solve larger instances, which performs well on large *Sym-G* instances and remarkably well on all *Rob-G* instances.

- Even if we devise them for the VPN design problem, the contents of this study, i.e., the linear MIP formulations and the solution algorithm, can be extended to the general multi-commodity network design problem with polyhedral demands.

6.1.2 Oblivious OSPF routing

Open Shortest Path First (OSPF) is a link-state routing protocol where the traffic demand is always routed on the corresponding shortest paths. Due to the 'length' restriction on the set of admissible paths, it is deemed to be inappropriate for traffic engineering. However, the recent weight management techniques enable to mitigate such criticisms to some extent. The motivation of a notable portion of research on OSPF is to have a comparable performance with MPLS, which is a more general routing mechanism with no restriction on the length of the paths. Polyhedral traffic uncertainty was initially used for the MPLS routing, and to the best of our knowledge, there is just one reference ([53]) on a rather restricted definition of feasible demands, which is based on a simulated annealing heuristic.

The purpose of our study on OSPF routing is to investigate if successful

traffic engineering could improve the OSPF performance for any arbitrary demand polyhedra. Our main contributions on this topic are as follows:

- For an arbitrary demand polyhedron, we provide a compact linear programming model for MPLS routing and hence show that an optimal oblivious MPLS routing can be computed in polynomial time.
- For an arbitrary demand polyhedron, we provide a compact mixed integer programming model for optimal oblivious OSPF routing with ECMP.
- We give an alternative MIP formulation for the OSPF routing problem and develop a branch-and-price algorithm strengthened with cuts as an exact solution tool for the two well-known demand polyhedra, namely the hose model and the robust variant we have also used for the robust VPN design problem.
- We provide a comparison of oblivious OSPF and oblivious MPLS routings for the two well-known polyhedral definitions.
- We discuss the improvement in OSPF performance through weight management as well as the impact of the demand uncertainty level on the oblivious OSPF and MPLS performances.

6.1.3 Network Loading Problem

Network Loading Problem (NLP) is a special type of the Capacitated Network Design Problem and it has received much attention in the literature. NLP admits variants according to the number of facility types and the number of commodities. The most frequently studied problems are the one or two facility cases with single commodity or multi-commodities. The common approach is to analyze the polyhedral structure of the set of feasible solutions to determine some strong valid inequalities so as to improve the LP relaxation bound. Unlike the single commodity problem, the multi-commodity variant remains very hard, and its solution requires the use of metric inequalities to define the projection of the

corresponding polyhedron on the space of discrete design variables. Moreover, to the best of our knowledge, there is no previous research incorporating uncertain demands in NLP other than Karaşan et al. [44] where uncertainty was included in the design of fiber optic networks with an emphasis on modeling rather than on a detailed polyhedral analysis and branch-and-cut.

We can summarize our main contributions as follows:

- We introduce a new type of NLP where we relax the assumption of known traffic demands and use polyhedral demand definition. We develop a compact linear mixed integer programming model for the problem.
- For an arbitrary number of facility types, we obtain an elegant property of the feasible set by projecting out the flow variables. We show that the existence of a multi-commodity flow can be certified by checking the existence of many single commodity flows.
- We provide a thorough polyhedral analysis of the problem for the hose model with multiple facilities.
- We present some valid inequalities for an arbitrary number of facilities and arbitrary capacity structures.
- We develop a branch-and-cut algorithm in which a simple but efficient heuristic is used to generate upper bounds. We observe that the algorithm performs exceptionally good for the single facility case and quite satisfactorily for the two facility case when compared with the commercial MIP solvers.

6.2 Future research directions

The evolutionary change that the telecommunication industry has gone through in the last couple of decades is an indicator of its importance for the information era. Introduction of new technologies as well as a more severe competition is

expected. Thus, the efficient and effective management of the telecommunication networks is likely to become the key issue. Therefore, the operations research community has to continue to produce new solution tools so as to keep up with the basic requirements of the industry. We believe that the network operators offering services agile and resistant to fluctuations in the working environment will have a competitive advantage. This was the main drive of this study. We aimed to improve current works and foster new research directions so as to encourage the design of robust networks. With this motivation, we investigated robust solutions to three well-known network design problems and made significant contributions.

We can emphasize several areas to perform further research so as to extend and improve the contributions we have made in this thesis. An important observation we need to make here is that the three robust problems we deal with are closely related. For a given set of Quallity of Service requirements, a network operator faces two major problems: to *dimension* the network capacity such that links can accommodate any feasible TM and to *route* the corresponding pairwise traffic demands by using network resources efficiently. Simply, the first task is related with the robust VPN design problem and NLP whereas the oblivious OSPF routing tackle the second part. As a result, we can suggest to unify the three studies in this thesis to provide an overall tool for the network operators.

Moreover, these three problems can be configured in a hierarchical structure. At the top level, there is the ISP, who provides its customers access to Internet and the related services. The problem of the ISP is to determine the capacity of the links in its private network such that it can offer a reliable service to its customers that is robust to fluctuations in demand estimates. The set of feasible demand realizations for the ISP network would be defined using the customer specific information. For example, detailed traffic estimates for existing customers would be available and hence box constraints in the BS model could be used for them. On the other hand, less restrictive conditions could be used for new customers. Therefore, our current results on robust NLP should be extended to tackle arbitrary demand polyhedra. On the other hand, we have the VPN customers in the second level of the hierarchy. The ISP faces the VPN design problem at this level to determine how to apportion the available capacity on the

links of its private network for the use of its VPN customers. Finally, given a link capacity configuraton for each VPN customer, the traffic engineering efforts for different measures of performance like congestion, call loss rate, average delay rate, jitter can be employed for the lower level of the hierarchy. We believe such a unified approach would not only improve the operational performance of the ISP provider but also increase customer satisfaction.

Another future research direction could be to study the VPN design problem with integer link capacities. Moreover, the computational complexity of the decision problem corresponding to the Sym-G case and the proof of the tree conjecture are some other interesting topics. In addition we also plan to consider the optimal routing problem where we backup for single link failures. Such an extension would surely enhance the operability of the network. Frankly, we believe that it is crucial to consider robustness within the survivable network design framework. However, it is definitely not so easy to accomplish.

In conclusion, we suggest to extend our current studies to different problems where the operating environment can be associated with a network. It will also be beneficial to use polyhedral uncertainty in conjunction with other design criteria depending on the needs of the system under consideration. Finally, a better analysis of the contemporary business environment and available technology can produce new demand polyhedra definitions whose use in the design of robust networks could be interesting.

Bibliography

[1] Instance originating from Dan Bienstock and provided by Pasquale Avella, 2004.

[2] E. Amaldi, P. Belotti, and M.Ç. Pınar. Multicommodity network design under polyhedral demand uncertainty. manuscript under preparation.

[3] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands:understanding fundamental tradeoffs. In *Proceedings of SIGCOMM*, pages 313–324, 2003.

[4] A. Atamtürk and O. Günlük. Network design arc set with variable upper bounds. *Networks*. to appear.

[5] A. Atamtürk and D. Rajan. On splittable and unsplittable capacitated network design arc-set polyhedra. *Mathematical Programming*, 92:315–333, 2002.

[6] P. Avella, S. Mattia, and A. Sassano. Metric inequalities and the network loading problem. *Discrete Optimization*, 4:103–114, 2007.

[7] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke. Optimal oblivious routing in polynomial time. In *Proceedings of STOC 03*, pages 383–388, San Diego, California, 2003.

[8] E. Balas and M. Oosten. On the dimension of projected polyhedra. *Discrete Applied Mathematics*, 87:1–9, 1998.

[9] C. Barnhart, C.A. Hane, and P.H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48:318–326, 2000.

[10] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46:316–329, 1998.

[11] P. Belotti and M.Ç. Pınar. Optimal oblivious routing under linear and nonlinear uncertainty. *Optimization and Engineering*, 2005. submitted for publication.

[12] W. Ben-Ameur and H. Kerivin. Routing of uncertain demands. *Optimization and Engineering*, 3:283–313, 2005.

[13] D. Berger, B. Gendron, J. Potvin, S. Raghavan, and P. Soriano. Tabu search for a network loading problem with multiple facilities. *Journal of Heuristics*, 6:253–267, 2000.

[14] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, Ser. B(98):43–71, 2003.

[15] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52:35–53, 2004.

[16] D. Bienstock, Sunil Chopra, O. Günlük, and Chih-Tang Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81:177–199, 1998.

[17] D. Bienstock and O. Günlük. Capacitated network design - polyhedral structure and computation. *INFORMS Journal on Computing*, 8:243–259, 1996.

[18] A. Bley and T. Koch. Integer programming approaches to access and backbone ip-network planning. Technical Report TR 02-41, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2002.

[19] P. Broström and K. Holmberg. Design of ip/ospf networks using a lagrangean heuristic on an in-graph based model. In *Proceedings of INOC*, volume B3, page 702, 2005.

[20] B. Burton, W. R. Pulleyblank, and Ph. L. Toint. *Lecture Notes in Economics and Mathematical Systems*, chapter The inverse shortest path problem with upper bounds on shortest path costs, pages 156–171. Springer, network optimization edition, 1997.

[21] A. Capone. personal communication, 2003.

[22] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J.E. van der Merive. A flexible model for resource management in virtual private networks. In *Proceedings of ACM SIGCOMM*, pages 95–108, 1999.

[23] T. Erlebach and M. Rüegg. Optimal bandwidth reservation in hose-model vpns with multi-path routing. In *Proceedings of IEEE Infocom*, volume 4, pages 2275–2282, March 2004.

[24] B. Fortz and M. Thorup. Internet traffic engineering by optimizing ospf weights. In *Proceedings of IEEE INFOCOM*, pages 519–528, 2000.

[25] B. Fortz and M. Thorup. Optimizing ospf/is-is weights in a changing world. *IEEE Journal on Selected Areas in Communications*, 20(4), May 2002.

[26] M. R. Garey and D. S. Johnson. *Computers and intractability: A guide to the theory of* NP*-completeness*. W.H. Freeman, San Francisco, 1979.

[27] L. De Giovanni, B. Fortz, and M. Labbé. A lower bound for the internet protocol network design problem. In *Proceedings of INOC*, volume B2, pages 401–408, 2005.

[28] O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86:17–39, 1999.

[29] O. Günlük, B. Brochmuller, and L. Wolsey. Designing private line networks - polyhedral analysis and computation. *Transactions on Operational Research*, 16:7–24, 2004.

[30] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network : A network design problem for multicommodity flow. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, number 389-398, Crete, Greece, 2001.

[31] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, pages 365–372, San Diego, CA, 2003.

[32] K. Holmberg and D. Yuan. Optimization of internet protocol network design and routing. *Networks*, 43(1):39–53, 2004.

[33] http://findvpn.com/articles/faq.php.

[34] http://searchsmb.techtarget.com.

[35] http://sndlib.zib.de/home.action.

[36] http://wikipedia.org.

[37] http://www.coin-or.org/documentation.html#BCP.

[38] C. Hurkens, J. Keijsper, and L. Stougie. Virtual private network design: a proof of the tree routing conjecture on ring networks. In M. Jünger and V. Kaibel, editors, *Proceedings of 11th International IPCO conference*, volume 3509 of *Lecture Notes in Computer Science series*, pages 40–421. Springer-Verlag, 2005.

[39] G. Italiano, R. Rastogi, and B. Yener. Restoration algorithms for virtual private networks in the hose model. *IEEE INFOCOM*, pages 131–139, 2002.

[40] G.F. Italiano, S. Leonardi, and G. Oriolo. Design of networks in the hose model. In *Proceedings of the* 3^{rd} *Workshop on Approximation and Randomization Algorithms in Communication Networks (ARANCE)*, pages 65–76, 2002.

[41] Kamal Jain, Ion Mandoiu, and Vijay Vazirani. A primal-dual schema based approximation algorithm for the element connectivity problem. In *Proceedings of Symposium on Discrete Algorithms*, 1999.

[42] V. Jiménez and A. Marzal. *Computing the K shortest paths: A new algorithm and an experimental comparison*, volume 1668 of *Lecture Notes in Computer Science series*, pages 15–29. Springer-Verlag, 1999.

[43] A. Jüttner, I.Szabó, and Á. Szentesi. On bandwidth efficiency of the hose resource management in virtual private networks. Infocom, IEEE, 2003.

[44] O. Karaşan, M. Ç. Pınar, and H. Yaman. Robust dwdm routing and provisioning under polyhedral demand uncertainty. 2006. *submitted for publication.*

[45] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for provisioning virtual private networks in the hose model. In *Proceedings of SIGCOMM*, pages 27–31, San Diego, California, USA, August 2001.

[46] M. Labbé and H. Yaman. Solving the hub location problem in a star-star network. *Networks.* to appear.

[47] F. Y. Lin and J. L. Wang. Minimax open shortest path first routing algorithms in networks supporting smds service. In *Proceedings of IEEE Int. Conf. Communications(ICC)*, volume 2, pages 666–670, 1993.

[48] T. L. Magnanti and P. Mirchandani. Shortest paths single origin-destination network design and associated polyhedra. *Networks*, 23:103–121, 1993.

[49] T. L. Magnanti, P. Mirchandani, and R. Vachani. The convex hull of two core capacitated network design problems. *Mathematical Programming*, 60:233–250, 1993.

[50] T. L. Magnanti, P. Mirchandani, and R. Vachani. Modeling and solving the two-facility capacitated network loading problem. *Operations Research*, 43(1):142–157, 1995.

[51] C. McDonald. Virtual private networks: An overview, 2004.

[52] P. Mirchandani. Projections of the network loading problem. *European Journal of Operational Research*, 122:534–560, 2000.

[53] E. Mulyana and U. Killat. Optimizing ip networks for uncertain demands using outbound traffic constraints. In *Proceedings of INOC*, 2005.

[54] K. Onaga and O. Kakusho. On feasibility conditions of multicommodity flows in networks. *IEEE Transactions on Circuit Theory*, 18(4):425–429, 1971.

[55] G. Oriolo. Domination in traffic matrices. Available for download at http://www.andrew.cmu.edu/user/belotti/papers/obl-stat.pdf, 2004.

[56] A. Parmar, S. Ahmed, and J. Sokol. An integer programming approach to the ospf weight setting problem. *submitted for publication*, 2006.

[57] M. Pióro, Á. Szentesi, J. Harmatos, A. Jüttner, P. Gajowniczek, and S. Kozdrowski. On open shortest path first related network optimization problems. *Performance Evaluation*, 48:201–223, 2002.

[58] R.L. Rardin and L.A. Wolsey. Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *European Journal of Operational Research*, 71:95–109, 1993.

[59] M.W.P. Savelsbergh, G.C. Sigismondi, and G.L. Nemhauser. A functional description of minto, a mixed integer optimizer. *OR Letters*, 15:47–58, 1994.

[60] A. Schrijver. *Theory of linear and integer programming.* John Wiley and Sons, New York, 1986.

[61] A.L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21:1154–1157, 1973.

[62] N. Springs, R. Mahajan, and D. Wetherall. Measuring isp topologies with rocketfuel. In *Proceedings of IEEE/ACM Transactions on Networking*, volume 12, pages 2–16, 2004.

[63] A. Sridharan, R. Guérin, and C. Diot. Achieving near-optimal traffic engineering solutions for current ospf/is-is networks. *IEEE INFOCOM*, 2003. San Francisco, CA.

[64] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. In *Proceedings of International Workshop on Approximation*

Algorithms for Combinatorial Optimization (APPROX), volume 2462 of *Lecture Notes in Computer Science series*, pages 256–270, 2002.

[65] A. Tomaszewski, M. Pióro, M. Dzida, and M. Zagożdżon. Optimization of administrative weights in ip networks using the branch-and-cut approach. In *Proceedings of INOC*, volume B2, pages 393–400, 2005.

[66] S. P.M. van Hoesel, A. M.C.A. Koster, R. L.M.J. van de Leensel, and M. W.P. Savelsbergh. Polyhedral results for the edge capacity polytope. *Mathematical Programming*, Ser. A(92):335–358, 2002.

[67] J. Wang, Y. Yang, L. Xiao, and K. Nahrstedt. Edge-based traffic engineering for ospf networks. *Computer Networks*. to appear.

[68] Y. Wang, Z. Wang, and L. Zhang. Internet traffic engineering without full mesh overlaying. In *Proceedings of IEEE INFOCOM*, pages 565–571, 2001.

[69] L. A. Wolsey. *Integer Programming*. Wiley-Interscience, NY, 1998.

[70] H. Yaman. The integer knapsack cover polyhedron. *SIAM Journal on Discrete Mathematics*. to appear.

[71] J. Zhang, Z. Ma, and C. Yang. A column generation method for inverse shortest path problems. *Mathematical Methods of Operations Research (ZOR)*, 41(3):347–358, 1995.

VITA

Ayşegül Altın was born on January 30, 1979 in Erzurum, Turkey. She received her high school education at Tekirdağ Anatolian High School. She earned her Bachelor of Science degree in Industrial Engineering from Marmara University in June 2001. She received a Master of Science degree in Management Science and Operational Research from Warwick University in September 2002. Afterwards, she has joined Bilkent University for her PhD study under the supervision of Prof. Mustafa Ç Pınar. Her main research interests are network design, integer programming, and robust optimization.

Starting from October 2007, she will join the Graphs and Mathematical Optimization Group in the Computer Science Department of Université Libre de Bruxelles as a post doc fellow.